

METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR ~~SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION~~SELECTING A RESOURCE BASED ON A MEASURE OF A PROCESSING COST

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 62/137,173 filed 3-23-2015 and is a continuation-in-part of, and claims priority to U.S. Patent Application No. 14/294,059, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SELECTING A RESOURCE BASED ON A MEASURE OF A PROCESSING COST," filed 06-02-2014 which in turn claims priority to U.S. Patent Application No. 12/857,851, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SELECTING A RESOURCE BASED ON A MEASURE OF A PROCESSING COST," filed 08-17-2010, which are each incorporated herein by

reference in their entirety for all purposes.

[0002] Additionally, this application is a continuation-in-part of, and claims priority to

U.S. Patent Application No. 13/941,502, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A NON-RETURNABLE COMMAND RESPONSE BASED ON A MARKUP ELEMENT," filed 07-14-2013 which in turn claims priority to U.S. Patent Application No. 12/789,550, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A NON-RETURNABLE COMMAND RESPONSE BASED ON A MARKUP ELEMENT," filed 05-

1

28-2010, which are each incorporated herein by reference in their entirety for all purposes.

[0003] Additionally, this application is a continuation-in-part of, and claims priority to

U.S. Patent Application No. 13/477,402, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION," filed 05-22-2012 which in turn claims priority to U.S. Patent Application No. 12/714,454-~~(Docket No 0120) filed on 2010/02/27, now US Patent No. ,~~ entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR

DETECTING AN IDLE TCP CONNECTION," filed 02-27-2010, which are each

incorporated herein by reference in their entirety for all purposes.

[~~0002~~0004] This application is related to the following ~~commonly owned U.S. Patent Applications, the entire disclosure~~ which are each ~~being~~ incorporated herein by reference ~~herein~~ in their entirety for all purposes: U.S. Patent Application No. 12/857,847, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PRESENTING AN INDICATION OF A COST OF PROCESSING A RESOURCE," filed 08-17-2010;

U.S. Patent Application No. 12/857,857, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SELECTING A RESOURCE IN RESPONSE TO A

CHANGE IN AVAILABLE ENERGY," filed on 08-17-2010; U.S. Patent Application No. 12/857,836, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR MAINTAINING A RESOURCE BASED ON A COST OF ENERGY," filed 08-17-2010;

U.S. Patent No. 8,346,853, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING AN ATTACHED COMMAND RESPONSE," filed 05-

2

27-2010; U.S. Patent Application No. 12/788,373, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PREVENTING PROCESSING OF AN HTTP RESPONSE," filed 05-27-2010; U.S. Patent Application No. 12/789,538, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING AN ATTACHED COMMAND RESPONSE BASED ON A MARKUP ELEMENT," filed 05-28-

2010; U.S. Patent Application No. 12/788,381, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A COMBINED COMMAND RESPONSE," filed 05-27-2010; U.S. Patent Application No. 12/789,568, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A COMBINED COMMAND RESPONSE BASED ON A MARKUP ELEMENT," filed 05-28-

2010; and U.S. Patent Application No. 12/714,063-(Docket No 0110)-filed on 2010/02/26, entitled "~~Methods, Systems, and Program Products for Detecting an Idle TCP Connection~~"METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR DETECTING AN IDLE TCP CONNECTION," filed on 02-26-2010.

BACKGROUND

~~{0003} Various implementations of the transmission control protocol (TCP) in network nodes support a number of options that are not negotiated or even communicated between or among any of the nodes. Some of these options are included in the specification of the TCP while others are not. For example, the TCP keep-alive option is supported by a number of implementations of the TCP. It is not, however, part of the TCP specification as described in "Request for Comments" (RFC) document RFC 793 edited by John Postel, titled "Transmission Control Protocol, DARPA Internet Program Internet Protocol Specification" (September 1981), which is incorporated here in its entirety by reference. One, both, or neither node including an endpoint in a TCP~~

~~connection may support a keep-alive option for the connection. Each node supports or does not support keep-alive for a TCP connection based on each~~

~~node's requirements without consideration for the other node in the TCP connection.~~

~~[0004] With respect to the keep-alive option, some argue that it is unnecessary and that it can waste network bandwidth. Some of these critics point out that a keep-alive packet can bring down a TCP connection. Further, since nodes including endpoints in a TCP connection do not cooperate in supporting the keep-alive option, the nodes may operate in opposition to one another and/or may waste resources by duplicating function, according to critics of the keep-alive option.~~

~~[0005] Proponents of the keep-alive option claim there is a benefit to detecting a dead peer/partner endpoint sooner. A node providing TCP keep-alive can also indirectly detect when a network is so congested that two nodes with endpoints in a TCP connection are effectively disconnected. Proponents argue that keep-alive can keep an inactive TCP connection open. For example, some network nodes such as firewalls are configured to close TCP connections determined to be idle or inactive in order to recover resources. Keep-alive can prevent this. This is good from the perspective of the node sending keep-alive packets, but the keep-alive packets might cause the firewall to waste resources and possibly block or terminate TCP connections with other nodes.~~

~~[0006] TCP keep-alive and the debate of its benefits and faults have been around for decades. To date no mechanism to allow two TCP connection endpoints to cooperate in supporting the keep-alive option has been proposed or implemented. The broader issue~~

~~of enabling cooperation and negotiation between nodes in a TCP connection in detecting and managing idle, underactive, and/or dead TCP connections remains unaddressed.~~

~~[0007]~~

[0005] Many personalization options on computing devices consume energy not required to perform computing tasks. For example, a primary purpose of a desktop background is esthetic. Mouse pointer effects, window effects, document previews, and many other examples exist. Currently, a user can configure a device to automatically turn such features on or off based on whether the device is plugged into an electrical

3

outlet or drawing energy from a battery. Other features, such as the brightness of a display, can be adjusted based on whether a device is plugged in or not. These configuration options give no consideration to resources being processed by the corresponding features.

[0006] Accordingly, there exists a need for methods, systems, and computer program products for ~~sharing information for detecting an idle TCP connection~~selecting a resource based on a measure of a processing cost.

SUMMARY

[~~0008~~0007] The following presents a simplified summary of the disclosure in order to provide a basic understanding to the reader. This summary is not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention. Its sole purpose is to present some concepts disclosed herein in a simplified form as a prelude to the more detailed description that is presented later.

[~~0009~~0008] Methods and systems are described for ~~sharing information for detecting an idle TCP connection~~selecting a resource based on a measure of a processing cost. In one aspect, ~~a~~the method includes receiving, ~~by a second node from a first node, a first transmission control protocol (TCP) packet in a TCP connection~~ resource information identifying a first resource and a second resource for processing by a program component. The method further includes ~~detecting a first idle time period header, in the first packet, identifying metadata for a first idle time period, detectable by the first node, during which no TCP packet including data in a first TCP data stream sent in the TCP connection by~~determining at least one of a first measure of a specified processing cost for the processing of the first resource and a second measure of the processing cost for the processing of the second ~~node is received by the first node~~resource. The method still further includes selecting one of the first resource and the second resource ~~modifying,~~ based on the ~~metadata, by~~at least one of the first measure and the second ~~node a timeout attribute associated with the TCP connection~~measure. The method still

4

further includes identifying, to the program component, the selected one of the first

resource and the second resource for processing.

[~~0010~~0009] Further, a system for ~~sharing information for detecting an idle TCP connection~~selecting a resource based on a measure of a processing cost is described. The system includes ~~an execution environment including an instruction processing unit configured to process an instruction included in at least one of a net in-port~~a cost advisor component, ~~an idle time period option handler~~a cost monitor component, ~~and an option attribute handler~~a cost director component, and a cost operations component adapted for operation in an execution environment. The system includes the ~~net in-port cost advisor~~ component configured for receiving, ~~by a second node from~~ resource information identifying a first ~~node, a first transmission control protocol (TCP) packet in a TCP connection~~resource and a second resource for processing by a program component. The system further includes the ~~idle time period option handler~~cost monitor component configured for ~~detecting~~determining at least one of a first idle time period header, in the first packet, ~~identifying metadata for a first idle time period, detectable by the first node, during which no TCP packet including data in a first TCP data stream sent in the TCP connection by~~measure of a specified processing cost for the processing of the first resource and a second measure of the processing cost for the processing of the second ~~node is received by the first node~~resource. The system still further includes the ~~option attribute handler~~cost director component configured ~~for modifying,~~

~~based on the metadata, by the second node a timeout attribute associated with the TCP connection~~

~~[0011] In another aspect, a method for sharing information for detecting an idle TCP connection is described that includes receiving, by a first node, first idle information for detecting a first idle time period during which no TCP packet including data in a first data stream sent in the TCP connection by a second node is received by the first node. The method further includes generating a TCP packet including a first idle time period header identifying metadata for the first idle time period based on the first idle information. The method still further includes sending the TCP packet in the TCP connection to the second node to provide the metadata for selecting one of the first idle time period to resource and the second node. The method also includes detecting the first idle time period resource based on the first idle information. The method additionally includes deactivating the TCP connection in response to detecting the first idle time period. the at least one of the first~~

~~[0012] Still further, a system for sharing information for detecting an idle TCP connection is described. The system includes an execution environment including an instruction processing unit configured to process an instruction included in at least one of an idle time period policy component, a packet generator component, a net out port component, an idle time period monitor component, and a connection state component. The system includes the idle time period policy component configured for receiving, by a first node, first idle information for detecting a first idle time period during which no TCP packet including data in a first data stream sent in the TCP connection by a second node is received by the first node. The system includes the packet generator component configured for generating a TCP packet including a first idle time period header identifying metadata for the first idle time period based on the first idle information. The system still further includes the net out port component configured for sending the TCP packet in the TCP connection to the second node to provide the metadata for the first idle time period to the second node. The system includes the idle time period monitor component configured for detecting the first idle time period based on the first idle information. The system includes the connection state component~~

~~configured for deactivating the TCP connection in response to detecting the first idle time period~~
measure and the second measure. The system still further includes the cost operations component configured for identifying, to the program component, the selected one of the first resource and the second resource for processing.

BRIEF DESCRIPTION OF THE DRAWINGS

~~[0013]~~[0010] Objects and advantages of the present invention will become apparent to those skilled in the art upon reading this description in conjunction with the accompanying drawings, in which like reference numerals have been used to designate like or analogous elements, and in which:

[~~0014~~0011] Fig. 1 is a block diagram illustrating an exemplary hardware device included in and/or otherwise providing an execution environment in which the subject matter may be implemented;

[0012] Fig. 2 is a flow diagram illustrating a method for selecting a resource based on a measure of a processing cost according to an aspect of the subject matter described herein;

[0013] Fig. 3 is a block diagram illustrating an arrangement of components for selecting a resource based on a measure of a processing cost according to another aspect of the subject matter described herein;

[0014] Fig. 4a is a block diagram illustrating an arrangement of components for selecting a resource based on a measure of a processing cost according to another aspect of the subject matter described herein;

[0015] Fig. ~~24b~~ is a ~~flowblock~~ diagram illustrating ~~a method for sharing information for detecting an idle TCP connection~~an arrangement of components for selecting a resource based on a measure of a processing cost according to ~~an~~another aspect of the subject matter described herein;

[0016] Fig. ~~34c~~ is a ~~flowblock~~ diagram illustrating ~~another method for sharing information for detecting an idle TCP connection~~an arrangement of components for selecting a resource based on a measure of a processing cost according to another aspect of the subject matter described herein;

[0017] Fig. 4d is a block ~~a~~-diagram illustrating an arrangement of components for ~~sharing information for detecting an idle TCP connection~~selecting a resource based on a measure of a processing cost according to ~~a further~~another aspect of the subject matter described herein;

6

[0018] Fig. 5 is a ~~block~~network diagram illustrating an ~~arrangement of components for sharing information for detecting an idle TCP connection~~exemplary system for selecting a resource based on a measure of a processing cost according to ~~still~~-another aspect of the subject matter described herein; and

[0019] Fig. 6 is a ~~network~~ diagram illustrating ~~an exemplary system for sharing information for detecting an idle TCP connection according to an aspect of the subject matter described herein;~~

~~{0020} Fig. 7 is a message flow diagram illustrating an exemplary data and execution flow for sharing information for detecting an idle TCP connection according to an aspect of the subject matter described herein; and~~

~~{0021} Fig. 8 is a diagram illustrating a structure for a packet transmitted a user interface presented via a network display according to an another aspect of the subject matter described herein.~~

DETAILED DESCRIPTION

[~~0022~~0020] One or more aspects of the disclosure are described with reference to the drawings, wherein like reference numerals are generally utilized to refer to like elements throughout, and wherein the various structures are not necessarily drawn to scale. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a

thorough understanding of one or more aspects of the disclosure. It may be evident, however, to one skilled in the art, that one or more aspects of the disclosure may be practiced with a lesser degree of these specific

details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing one or more aspects of the disclosure. [0021] An exemplary device included in an execution environment that may be configured according to the subject matter is illustrated in Fig. 1. An execution environment includes an arrangement of hardware and, optionally, software that may be further configured to include an arrangement of components for performing a method of the subject matter described herein.

~~{0023}~~ An execution environment includes and/or is otherwise provided by one or more devices. An execution environment may include a

7

virtual execution environment including software components operating in a host execution environment. Exemplary devices included in or otherwise providing suitable execution environments for configuring according to the subject matter include personal

computers, notebook computers, tablet computers, servers, hand-held and other mobile devices, multiprocessor devices, distributed devices, consumer electronic devices, routers, communication servers, and/or other network-enabled devices. Those skilled in the art will understand that the components illustrated in Fig. 1 are exemplary and may vary by particular execution environment.

~~[0024]~~[0022] Fig. 1 illustrates hardware device 100 included in execution environment 102 ~~which. Fig. 1 illustrates that execution environment 102~~ includes instruction-processing unit (IPU) 104, such as one or more microprocessors; physical IPU memory 106 including storage locations identified by addresses in a physical memory address space of IPU 104; persistent secondary storage 108, such as one or more hard drives and/or flash storage media; input device adapter 110, such as a key or keypad hardware, a keyboard adapter, and/or a mouse adapter; output device adapter 112, such as a display and/or an audio adapter for presenting information to a user; a network interface component, illustrated by network interface adapter 114, for communicating via a network such as a LAN and/or WAN; and a communication mechanism that couples elements 104-114, illustrated as bus 116. Elements 104-114 may be operatively coupled by various means. Bus 116 may comprise any type of bus architecture, including a memory bus, a peripheral bus, a local bus, and/or a switching fabric.

~~{0025} IPU 104 is an instruction execution machine, apparatus, or device. Exemplary IPUs include one or more microprocessors, digital signal processors (DSP), graphics processing units (GPU), application-specific integrated circuits (ASIC), and/or field programmable gate arrays (FPGA).~~

~~{0026} IPU 104 may access machine code instructions and data via one or more memory address spaces in addition to the physical memory address space. A memory address space includes addresses identifying locations in an IPU~~

~~memory. IPU 104 may have more than one IPU memory. Thus, IPU 104 may have more than one memory address space. IPU 104 may access a location in an IPU memory by processing an address identifying the location. The processed address may be in an operand of a machine code instruction and/or may be identified in a register or other portion of IPU 104.~~

~~[0027] Fig. 1 illustrates virtual IPU memory 118 spanning at least part of physical IPU memory 106 and at least part of persistent secondary storage 108. Virtual memory addresses in a memory address space may be mapped to physical memory addresses identifying locations in physical IPU memory 106. An address space for identifying locations in a virtual IPU memory is referred to as a virtual memory address space; its addresses are referred to as virtual memory addresses; and its IPU~~

8

~~processor~~ memory ~~is known as a virtual IPU memory or virtual memory. The term IPU memory may refer to physical IPU memory 106 and/or, such as~~ virtual IPU memory 118, depending on the context in which the term is used.

~~[0028] Various Physical IPU memory 106 may include various types of memory technologies may be included in physical IPU memory~~

~~106.~~ Exemplary memory technologies include static random access memory (SRAM) and/or dynamic RAM (DRAM) including variants such as dual data rate synchronous DRAM (DDR SDRAM), error correcting code synchronous DRAM (ECC SDRAM), and/or RAMBUS DRAM (RDRAM). Physical IPU memory 106 may include volatile memory as illustrated in the previous sentence and/or may include nonvolatile memory such as nonvolatile flash RAM (NVRAM) and/or ~~read-only memory (ROM).~~

~~[0029] Persistent secondary storage 108 may include one or more flash memory storage devices, one or more hard disk drives, one or more magnetic disk drives, and/or one or more optical disk drives. Persistent secondary storage may include removable media. The drives and their associated computer-readable storage media provide volatile and/or nonvolatile storage for computer readable instructions, data structures, program components, and other data for execution environment 102.~~

~~[0030] Execution environment 102 may include software components stored in~~

~~persistent secondary storage 108, in remote storage accessible via a network, and/or in an IPUa processor memory. Fig. 1 illustrates execution environment 102 including operating system 120, one or more applications 122, and other program code and/or data components illustrated by other libraries and subsystems 124. In an aspect, some or all software components may be stored in locations accessible to IPU 104 in a shared memory address space shared by the software components. The software components~~

~~+0031~~

10

accessed via the shared memory address space are stored in a shared processor memory defined by the shared memory address space. In another aspect, a first software component may be stored in one or more locations accessed by IPU 104 in a first address space and a second software component may be stored in one or more locations accessed by IPU 104 in a second address space. The first software component is stored in a first processor memory defined by the first address space and the second software component is stored in a second processor memory defined by the second address space.

[0028] Software components typically include instructions executed by IPU 104 in a computing context referred to as a "process". A process may include one or more "threads". A "thread" includes a sequence of instructions executed by IPU 104 in a computing sub-context of a process. The terms "thread" and "process" may be used interchangeably herein when a process includes only one thread.

[0029] Execution environment 102 may receive user-provided information via one or more input devices illustrated by input device 128. Input device 128 provides input information to other components in execution environment 102 via input device adapter

110. Execution environment 102 may include an input device adapter for a keyboard, a touch screen, a microphone, a joystick, a television receiver, a video camera, a still camera, a document scanner, a fax, a phone, a modem, a network interface adapter, and/or a pointing device, to name a few exemplary input devices.

~~[0032]~~0030 Input device 128 included in execution environment 102 may be included in device 100 as Fig. 1 illustrates or may be external (not shown) to device 100. Execution

11

environment 102 may include one or more internal and/or external input devices. External input devices may be connected to device 100 via corresponding communication interfaces such as a serial port, a parallel port, and/or a universal serial bus (USB) port. Input device adapter 110 receives input and provides a representation to bus 116 to be received by IPU 104, physical IPU memory 106, and/or other components included in execution environment 102.

~~[0033]~~0031 Output device 130 in Fig. 1 exemplifies one or more output devices that may be included in and/or may be external to and operatively coupled to device 100. For example, output device 130 is illustrated connected to bus 116 via output device adapter 112. Output device 130 may be a display device. Exemplary display devices include liquid crystal displays (LCDs), light emitting diode (LED) displays, and projectors. Output device 130 presents output of execution environment 102 to one or more users. In some embodiments, an input device may also include an output device ~~is a device such as~~. Examples include a phone, a joystick, and/or a touch screen. In addition to various types of display devices, exemplary output devices include printers, speakers, tactile output devices such as motion producing devices, and other output devices producing sensory information detectable by a user.

~~[0034]~~0032 A device included in or otherwise providing an execution environment may operate in a networked environment communicating with one or more devices ~~(not shown)~~ via one or more network ~~interfaces~~interface components. The terms "communication interface component" and "network

interface component" are used interchangeably. Fig. 1 illustrates network interface adapter (NIA) 114 as a network interface component
12

included in execution environment 102 to operatively couple device 100 to a network. ~~The terms "A network node" and "node" in this document both refer to a device having~~interface component includes a network interface ~~operatively coupled to a network~~hardware (NIH) component and optionally a software component.

[~~0035~~0033] Exemplary network ~~interfaces~~interface components include ~~wireless network~~network interface controller components, network interface cards, network interface adapters, and line cards. A node may include one or more network interface components to interoperate with a wired network ~~adapters~~and/or a wireless network. Exemplary wireless networks include a BLUETOOTH network, a wireless 802.11 network, and/or a wireless telephony network (e.g., a cellular, PCS, CDMA, and/or GSM network). Exemplary ~~wired networks include various types of LANs, wide area networks (WANs), and personal area networks (PANs).~~Exemplary network adaptersinterface components for wired networks include Ethernet adapters, Token-ring adapters, FDDI adapters, asynchronous transfer mode (ATM) adapters, and modems of various types. Exemplary wired and/or wireless networks include various types of LANs, WANs, and/or personal area networks (PANs). Exemplary networks also include intranets and internets such as the Internet.

~~{0036} Fig. 2 is a flow diagram illustrating a first method for sharing information for detecting an idle TCP connection according to an exemplary aspect of the subject matter described herein. Fig. 3 is a flow diagram illustrating a second method for sharing information for detecting an idle TCP connection according to an exemplary~~

~~aspect of the subject matter described herein. Fig. 4a is a block diagram illustrating a system for sharing information for detecting an idle TCP connection according to the first method in Fig. 2. Fig. 4b is a block diagram illustrating a system for sharing information for detecting an idle TCP connection according to the second method in Fig.~~

~~3. It is expected that many, if not most, systems configured to perform one of the methods illustrated in Fig. 2 and Fig. 3 will also be configured to perform the other method.~~

~~{0037} A system for sharing information for detecting an idle TCP connection~~

[0034] The terms "network node" and "node" in this document both refer to a device having a network interface component for operatively coupling the device to a network. Further, the terms "device" and "node" used herein refer to one or more devices and nodes, respectively, providing and/or otherwise included in an execution environment unless clearly indicated otherwise.

[0035] The components of a user interface are generically referred to herein as "user interface elements". More specifically, visual components of a user interface are referred to herein as "visual interface elements". A visual interface element may be a

13

visual component of a graphical user interface (GUI). Exemplary visual interface elements include windows, textboxes, sliders, list boxes, drop-down lists, spinners, various types of menus, toolbars, ribbons, combo boxes, tree views, grid views, navigation tabs, scrollbars, labels, tooltips, text in various fonts, balloons, dialog boxes, and various types of button controls including check boxes and radio buttons. An application interface may include one or more of the elements listed. Those skilled in the art will understand that this list is not exhaustive. The terms "visual representation", "visual component", and "visual interface element" are used interchangeably in this document. Other types of user interface elements include audio output components referred to as "audio interface elements", tactile output components referred to as "tactile interface elements", and the like.

[0036] A "user interface (UI) element handler" component, as the term is used in this document, includes a component configured to send information representing a program entity for presenting a user detectable representation of the program entity by an output device, such as a display. A "program entity" is an object included in and/or otherwise processed by an application or executable. The user detectable representation is presented based on the sent information. The sent information is referred to herein as "presentation information". Presentation information may include data in one or more formats. Exemplary formats include image formats such as JPEG, video formats such as MP4, markup language data such as HTML and other XML-based markup, and/or instructions such as those defined by various script languages, byte code, and/or machine code. For example, a web page received by a browser from

14

a remote application provider may include HTML, ECMAScript, and/or byte code for presenting one or more user interface elements included in a user interface of the remote application. Components configured to send information representing one or more program entities for presenting particular types of output by particular types of output devices include visual interface elements, audio interface element handler components, tactile interface element handler components, and the like.

[0037] A representation of a program entity may be stored and/or otherwise maintained in a presentation space. As used in this document, the term "presentation space" refers to a storage region allocated and/or otherwise provided for storing presentation information, which may include audio, visual, tactile, and/or other sensory data for presentation by and/or on an output device. For example, a buffer for storing an image and/or text string may be a presentation space. A presentation space may be physically and/or logically contiguous or non-contiguous. A presentation space may have a virtual as well as a physical representation. A presentation space may include a storage location in processor memory, secondary storage, a memory of an output adapter device, and/or a storage medium of an output device. A screen of a display, for example, is a presentation space.

[0038] As used herein, the term "program" or "executable" refers to any data representation that may be translated into a set of machine code instructions and optionally associated program data. Thus, a program or executable may include an application, a shared or non-shared library, and a system command.

Program representations other than machine code include object code, byte code, and source
15

code. Object code includes a set of instructions and/or data elements that either are prepared for linking prior to loading or are loaded into an execution environment. When in an execution environment, object code may include references resolved by a linker and/or may include one or more unresolved references. The context in which this term is used will make clear that state of the object code when it is relevant. This definition can include machine code and virtual machine code, such as Java™ byte code. [0039] As used herein, an "addressable entity" is a portion of a program, specifiable in programming language in source code. An addressable entity is addressable in a program component translated for a compatible execution environment from the source code. Examples of addressable entities include variables, constants, functions, subroutines, procedures, modules, methods, classes, objects, code blocks, and labeled instructions. A code block includes one or more instructions in a given scope specified in a programming language. An addressable entity may include a value. In some places in this document "addressable entity" refers to a value of an addressable entity. In these cases, the context will clearly indicate that the value is being referenced.

[0040] Addressable entities may be written in and/or translated to a number of different programming languages and/or representation languages, respectively. An addressable entity may be specified in and/or translated into source code, object code, machine code, byte code, and/or any intermediate languages for processing by an interpreter, compiler, linker, loader, or analogous tool.

[0041] The block diagram in Fig. 3 illustrates an exemplary system for selecting a resource based on a measure of a processing cost according to the method illustrated

16

~~in Fig. 2 includes. Fig. 3 illustrates a system, adapted for operation in an execution environment, such as execution environment 102 in Fig. 1, including an instruction processing unit, such as IPU 104, configured to process an instruction included in at least one of an idle time period policy component 450, a packet generator component 452, and a net out port component 454, an idle time period monitor component 456, and a connection state component 458 illustrated in Fig. 4a.~~

~~{0038} A system for sharing information for detecting an idle TCP connection performing the method illustrated in Fig. 3. The system illustrated includes a cost advisor component 302, a cost monitor component 304, a cost director component 306, and a cost operations component 308. The execution environment, such as execution environment 102 in Fig. 1, including includes an instruction processing unit, such as IPU 104, configured to process for processing an instruction included in at least one of a net in port the cost advisor component 460 302, an idle time period option handler the cost monitor component 462 304, an option attribute handler the cost director component 464 illustrated in Fig. 4b.~~

~~{0039} Components 306, and the cost operations component 308. Some or all of the exemplary components~~ illustrated in Fig. 4a~~3~~ may be adapted for performing the method illustrated in Fig. 2 in a number of execution environments. ~~Components illustrated in Fig. 4b may be adapted for performing the method illustrated in Fig. 3 in a number of execution environments. Fig. 5 is a~~ Figs. 4a-d include block diagram diagrams illustrating adaptations the components of Fig. 3 and/or analogs of the components of Fig. 4a and Fig. 4b~~3~~ adapted for operation in ~~exemplary various execution environment 502~~

environments 401 including or otherwise provided by one or more nodes. ~~The method depicted in~~

~~[0042] Fig. 2 and the method depicted in Fig. 3 may be carried out by some or all of the exemplary components and/or their analogs~~1 illustrates components of an exemplary device that may at least partially provide and/or otherwise be included in an execution environment.

~~{0040} The components illustrated in FigFigs. 4 and Fig. 5a-d may be included in or otherwise may be combined with some or all of the components of Fig. 1 to create a variety of arrangements of components according to the subject~~

matter described herein.

~~[00410043] Fig. 65 illustrates firstuser node 602 and second node 604502 as an~~ exemplary devicesdevice that in various aspects may be included in and/or otherwise adapted for providing ~~a suitable~~any of execution ~~environment, such as execution environment 502~~environments 401 illustrated in FigFigs. 5, ~~for an adaptation of the arrangement of components in Fig. 4a and an 4a-c each illustrating a different~~ adaptation of the ~~arrangement of components in Fig. 4b. As illustrated in Fig. 6, first node 602 and second node 604 are operatively coupled to network 606 via respective network interfaces enabling first node 602 and second node 604 to communicate. Fig. 7 is a message flow diagram illustrating an exemplary exchange of messages within and between first node 602 and second node 604 according to the subject matter described herein.~~

~~{0042} As stated, the various adaptations of the arrangements of components in Fig. 4a and in Fig. 4b described herein are not exhaustive.~~

~~{0043} In Fig. 5,~~ arrangement of components in Fig. 3. As illustrated in Fig. 5, user node 502 is operatively coupled to network 504 via a network interface component, such as network

17

interface adapter 114. Alternatively or additionally, an adaptation of an execution environment 502 illustrates a network401 may include and/or may otherwise be provided by a device that is not operatively coupled to a network. A server device is illustrated by application 504 operating in aprovider node configured to communicate with one or more other nodes via the TCP supported by TCP layer component 506. For example, first506. Application provider node

~~602506~~ may be included in and/or ~~provide~~ otherwise adapted for providing execution environment ~~502~~. ~~Network 401d~~ illustrated in Fig. 4d. As illustrated in Fig. 5, application provider node 506 is operatively coupled to network 504 ~~may be a first application configured to communicate with an application operating in second node 604 via a network 606. Second node 604 may be interface component~~ included in and/or ~~provide another instance of execution environment 502. The operation of both first node 602 and second node 604 are described with respect to 401d.~~

[0044] Fig. 4a illustrates execution environment 502. ~~For ease of illustration, both first node 602 and second node 604 are configured with adaptations~~ 401a hosting application 403a including an adaptation of the arrangement of components in Fig. 3. Fig. 4a and b illustrates execution environment 401b hosting browser 403b including an adaptation of the arrangement in Fig. 4b. As such, the description of components and corresponding operations with respect to execution environment 502 in Fig. 5 is applicable to both first node 602 and second node 604 in Fig. 6.

~~{0044} In Fig. 5, in Fig. 3 that may operate at least partially in a network interface card (NIC) 508 is an exemplification of a~~ application agent 405b received from a remote application provider, such as network interface illustrated application 403d in Fig. 1 by network interface adapter 114. NIC 508 includes a physical layer component 510 operatively coupling execution environment 502 to one or more physical media for carrying communication signals. The media ~~4d. Browser 403b and execution environment 401b may provide at least part of an execution environment for network application agent 405b that may be wired, such as an Ethernet LAN operating over CAT 6 cabling, or may be wireless such as an 802.11n LAN. Other exemplary physical layer protocols and corresponding media are identified above.~~

~~{0045} NIC 508 may also include a portion of link layer component 512. Link layer component 512 may provide for communication between two nodes in a point to point communication and/or two nodes in a local area network (LAN). Exemplary link layers and, their protocols have been described above including FDDI, ATM, and Ethernet. A portion of link layer component 512 is external to NIC 508. The external portion may be realized as a device driver for NIC 508.~~

~~{0046} Link layer component 512 may receive data formatted as one or more internet protocol (IP) packets from internet protocol (IP) layer component 514. Link layer component 512 packages data from IP layer component 514 according to the particular link layer protocol supported. Analogously, link layer component 512 interprets data, received as signals transmitted by the physical media operatively coupled to physical layer component 510, according to a particular link layer protocol supported. Link layer component 512 may strip off link layer specific data and transfer the payload of link layer transmissions to IP layer component 514.~~

~~{0047} IP layer component 514 illustrated in Fig. 5 is configured to communicate with one or more remote nodes over a LAN and/or a network of networks such as an intranet or the Internet. IP layer component 514 may receive data formatted as TCP packets from TCP layer component 506. IP layer component 514 packages data from TCP layer component 506 into IP packets for transmission across a network. The network may be and/or may include an~~

~~internet. Analogously, IP layer component 514 interprets data, received from link layer component 512 as IP protocol data and detects IP packets in~~

~~the received data. IP layer component 514 may strip off IP layer specific data and transfer the payload of one or more IP packets to TCP layer component 506.~~

~~[0048] In Fig. 5, IP layer component 514 is operatively coupled to TCP layer component 506. TCP layer component 506 is configured to provide a TCP connection over network 606 for sending and/or receiving packets included in the TCP connection between two nodes exemplified by first node 602 and second node 604.~~

~~[0049] In a TCP connection including first node 602 and second node 604, first node 602 may include a first TCP connection endpoint and second node 604 may include a second TCP connection endpoint. The first and second TCP connection endpoints identify the TCP connection. The TCP connection may have other identifiers, in addition to the included endpoints.~~

~~[0050] Components of execution environment 502, in an aspect, may interoperate with TCP layer~~received via a network from a network application operating in a remote execution environment. Fig. 4c illustrates an arrangement of the components in Fig. 3 adapted to operate in a cost management subsystem 407c of execution environment 401c. The arrangement in Fig. 4c may mediate communication between applications 403c and one or more output devices, such as output device 130 in Fig 1.

[0045] Fig. 4d illustrates execution environment 401d configured to host one or more network applications, such as a web service, illustrated by network application 403d. Fig. 4d also illustrates network application platform 409d that may provide services to

18

one or more network applications. Network application 403d includes yet another

adaptation of the arrangement of components in Fig. 3.

[0046] The various adaptations of the arrangement in Fig. 3 that are described herein are not exhaustive. For example, those skilled in the art will see based on the description herein that arrangements of components for performing the method illustrated in Fig. 2 may be at least partially included in an application and at least

partially external to the application. Further, arrangements for performing the method illustrated in Fig. 2 may be distributed across more than one node and/or execution environment. For example, such an arrangement may operate at least partially in browser 403b in Fig. 4b and at least partially in execution environment 401d in and/or external to network application 403d.

[0047] Figs. 4a-d illustrate adaptations of network stacks 411 configured for sending and receiving messages over a network, such as network 504, via a network interface component~~506 directly. In another aspect,~~Network application platform 409d in Fig. 4d provides services to one or more~~components~~network applications. In various aspects, such as~~network application 504,~~platform 409d may include and/or interoperate with TCP layer

~~component 506 indirectly. Network application 504 may exchange data with TCP layer component 506 via sockets component 518 and/or an analog of sockets component 518. Alternatively or additionally, network application 504a web server. Fig. 4d also illustrates network application platform 409d configured for interoperating with network stack 411d.~~

[0048] Network stacks 411 may support the same protocol suite, such as TCP/IP, or may communicate via a network gateway or other protocol translation device and/or service. For example, browser 403b in Fig. 4b and network application platform 409d in

Fig. 4d may interoperate via their respective network stacks: network stack 411b and network stack 411d.

19

[0049] Figs. 4a-d illustrate applications 403, respectively, which may communicate ~~with a remote node~~ via ~~anone or more~~ application ~~protocol~~ layer ~~illustrated by~~ protocols. Figs. 4a-d respectively illustrate application protocol ~~layer component 520. Many components 413 for communicating via one or more~~ application ~~layer~~ protocols ~~currently exist and new application protocols will be developed~~. Exemplary application ~~layer~~ protocols include hypertext transfer protocol (HTTP), ~~file transfer protocol (FTP), and extensible instant~~ messaging and presence (XMPP-IM) protocol ~~(XMPP).~~

~~{0051} TCP layer component 506 in Fig. 5. Matching protocols enabling applications 403 to communicate via network 504 in Fig. 5 are not required, if communication is via a protocol gateway or other translator.~~

[0050] In Fig. 4b, browser 403b may receive data from any of various sources for transmitting in corresponding TCP connections to various corresponding identified TCP connection endpoints, some or all of network application agent 405b in one or more messages sent from a network nodes. Fig. 5 illustrates application in port (app in port) component 522 providing an interface, such as network application 403d via network application platform 409d, a network stack 411, a network interface component, and optionally an application protocol component for receiving data to transmit in a TCP connection 413. In Fig. 5 illustrates TCP layer component 506 4b, browser 403b includes packet generator/content manager component 552 configured to package data received by application in port 415b. Content manager component 522 for transmitting in 415b may interoperate with one or more TCP packets. The one or more TCP packets are provided to IP layer component 514 via net out port component 554 exemplifying an output interface component.

of application protocol components 413b and/or network stack 411b to receive the message or messages including some or all of network application agent 405b.

[0051] Network application agent 405b may include a web page for presenting a user interface for network application 403d. The web page may include and/or reference data represented in one or more formats including hypertext markup language (HTML) and/or other markup language, ECMAScript or other scripting language, byte code, image data, audio data, and/or machine code.

[0052] Analogously, TCP layer component 506 interprets data. In an example, in response to a request received from IP layer browser 403b, controller component 514 via net in port component 560. The data is interpreted as TCP data and TCP packets are detected in the received data by net in port component 560 and/or packet handler component 516. Fig. 5 illustrates TCP layer component

~~506 includes packet handler component 516 to strip off and/or otherwise process TCP layer specific data. Packet handler component 516 interoperates with application out port (app out port) component 524 to transfer data in the TCP packet included in a TCP data stream to sockets component 518, application protocol layer 520, network application 504, and/or other components associated with the local endpoint of the TCP connection. Detailed information on the operation of the TCP is included in RFC 793.~~

~~[0053]~~ 417d, in Fig. 4d, may invoke model subsystem 419d to perform request-
20

specific processing. Model subsystem 419d may include any number of request handlers (not shown) for dynamically generating data and/or retrieving data from model database 421d based on the request. Controller component 417d may further invoke template engine 423d to identify one or more templates and/or static data elements for generating a user interface for representing a response to the received request. Fig. 4d

illustrates template database 425d including exemplary template 427d. Fig. 4d illustrates template engine 423d as a component in view subsystem 429d configured to return responses to processed requests in a presentation format suitable for a client, such as browser 403b. View subsystem 429d may provide the presentation data to controller component 417d to send to browser 403b in response to the request received from browser 403b. Some or all of network application agent 405b may be sent to browser 403b via network application platform 409d as described above.

[0053] While the example describes sending some or all of network application agent 405b in response to a request, network application 403d additionally or alternatively may send some or all of a network application agent to browser 403b via one or more asynchronous messages. In an aspect, an asynchronous message may be sent in response to a change detected by network application 403d. Publish-subscribe protocols, such as the presence protocol specified by XMPP-IM, are exemplary protocols for sending messages asynchronously.

[0054] The one or more messages including information representing some or all of network application agent 405b in Fig. 4b may be received by content manager component 415b via one or more of application protocol component(s) 413b and

21

network stack 411b as described above. In Fig. 4b, browser 403b includes one or more content handler components 431b to process received data according to its data type, typically identified by a MIME-type identifier. Exemplary content handler components 431b include a text/html content handler component for processing HTML documents;

an application/xmpp-xml content handler component for processing XMPP streams including presence tuples, instant messages, and publish-subscribe data as defined by various XMPP specifications; one or more video content handler components for processing video streams of various types; and still image data content handler components for processing various images types. Content handler components 431b process received data and may provide a representation

of the processed data to one or more user interface (UI) element handler components 433b.

[0055] UI element handler components 433 are respectively illustrated in presentation controller components 435 in Fig. 4a, Fig. 4b, and Fig. 4c. A presentation controller component 435 may manage visual, audio, and/or other types of output of its including application 403 as well as receive and route detected user and other inputs to components and extensions of its including application 403. With respect to Fig. 4b, a UI element handler component 433b in various aspects may be adapted to operate at least partially in a content handler component 431b such as a text/html content handler component and/or a script content handler component. Additionally or alternatively, a UI element handler component 433 in an execution environment 401 may operate in and/or as an extension of its including application 403. For example, a plug-in may provide a virtual machine, for a UI element handler component received as a script

22

and/or byte code, that may operate as an extension in application 403 and/or external to and interoperating with application 403.

[0056] Fig. 6 illustrates display presentation space 602 of a display in and/or operatively coupled to user node 502. Fig. 6 illustrates desktop background 604 that may be a still image and/or a video background. Selection window 606 is illustrated including selectable resource icons 608. In an aspect, a resource icon may represent image and/or video data. Resource icon 6082b is illustrated as selected. A selected image file and/or video stream that corresponds to a selected resource icon 608 may be processed in response to user input corresponding to operations illustrated in operation bar 610. Selection window 606 may be a user interface presented by any of applications 403 illustrated in Figs. 4a-d and/or by network application agent 405b. For example, selection window 606 may be presented via interoperation of browser 403b, network application agent 405b, and/or network application 403d. A browser window may include a user interface of a network application provided by a remote node, such as a network application 403d in Fig. 4d.

[0057] Various UI elements of applications 403 described above may be presented by one or more UI element handler components 433 in Figs. 4a-c and/or by one or more template engines 423d in Fig. 4d. In an aspect, illustrated in Figs. 4a-4c, UI element handler component(s) 433 of one or more applications 403 is/are configured to send representation information representing a visual interface element, such as operation bar 610 in Fig. 6, to a GUI subsystem 437. A GUI subsystem 437 may instruct a graphics subsystem 439 to draw the visual interface element in a region of display

23

presentation space 602, based on presentation information received from a corresponding UI element handler component 433.

[0058] Input may be received corresponding to a UI element via an input driver 441 illustrated in Figs. 4a-c in various adaptations. For example, a user may move a mouse to move a pointer presented in a display presentation space 602

over an operation user interface element presented in an operation bar 610. A user may provide an input detected by the mouse. The detected input may be received by a GUI subsystem 437 via an input driver 441 as an operation or command indicator based on the association of the shared location of the pointer and the operation user interface element in display presentation space 602.

[0059] With reference to ~~the method illustrated in~~ Fig. 2, block 202 illustrates that the method includes receiving, ~~by a first node, first idle resource~~ information ~~for detecting/identifying~~ a first ~~idle time period during which no TCP packet including data in a first data stream sent in a TCP connection by a second node is received by the first node~~resource and a second resource for processing by a program component. Accordingly, a system for ~~sharing information for detecting an idle TCP connection~~selecting a resource based on a measure of a processing cost includes means for receiving, ~~by a first node, first idle resource~~ information ~~for detecting/identifying~~ a first ~~idle time period during which no TCP packet including data in a first data stream sent in a TCP connection by a second node is received by the first node~~resource and a second resource for processing by a program component. For example, as illustrated in Fig. 4a, ~~idle time period policy component 450~~3, cost advisor component 302 is configured for receiving, ~~by a first node, first idle resource~~ information ~~for detecting a first idle time period during which no TCP packet including data in a first data stream sent in a TCP connection by a second node is received by the first node~~identifying a first resource and a second resource for processing by a program component. Figs. 4a-d illustrate cost advisor

~~{0054} Fig. 5 illustrates idle time period (ITP) policy component 550 as an adaptation of components 402 as adaptations and/or analog/analogues of ITP policy cost advisor component 450~~302 in Fig. 4a. ~~One or more idle time period policy components 550 operate in execution environment 502.~~

~~{0055} Message 702 in Fig. 7 illustrates a communication including and/or otherwise identifying idle information received by ITP policy component 550. Message 702 may take various forms in various aspects. Exemplary forms for message 702 include a function/method invocation, a message passed via a message queue, data transmitted via a pipe, a message received via a network, and/or a communication via a shared location in IPU memory and/or secondary storage.~~

~~{0056} Idle information may be received from a configuration storage location for TCP layer component 506 in an IPU memory and/or in secondary storage 108. The~~

~~configured idle information may be maintained and/or otherwise managed by settings service component 526 configured to maintain and/or manage various options or settings for TCP layer component 506 and/or one or more TCP connections.~~

~~{0057} In an aspect, network application 504 provides idle information to ITP policy component 550 via settings service component 526 interoperating with sockets component 518. Sockets component 518 and/or TCP layer component 506 may support TCP options applicable globally for some or all TCP connections and/or may support TCP options on a per connection basis. Per connection TCP~~

~~options may override global TCP options if global options are also supported. In another aspect, idle information may be received from and/or otherwise received based on information via application protocol layer 520, via sockets component 518, and/or directly from network application 504.~~

~~{0058} Application protocol layer 520 may provide idle information to ITP policy component 550 via settings service component 526 and, optionally, via sockets component 518. Idle information provided by application protocol layer 520 may be based on data received from network application 504, based on a particular configuration of application protocol layer 520, and/or received from a user and/or administrator of one or both of network application 504 and application protocol layer 520.~~

~~{0059} In an aspect, the idle information received may be based on a previous ITP header identified in a packet in the TCP connection received by first node 602 from~~

~~second node 604. The previous packet may be received by net in-port component 560. The previous ITP header may be detected by ITP option handler component 562 interoperating with packet handler component 516. Idle information may be identified and/or otherwise determined by ITP option handler component 562. ITP policy component 550 may interoperate with ITP option handler component 562 to receive the idle information.~~

~~{0060} Idle information received, determined, and/or otherwise identified may include and/or identify a duration of time for detecting an idle time period. The duration may be specified according to various measures of time including seconds, minutes, hours, and/or days.~~

~~{0061} Alternatively or additionally, idle information may include and/or identify a generator for determining a duration of time for detecting an idle time period. An exemplary generator may include a formula, an expression, a function, a policy, and/or other mechanism for generating and/or otherwise identifying a duration of time.~~

~~{0062} In an aspect, one or more algorithms for generating a duration of time for detecting an idle time period may be associated with identifiers. The algorithm identifiers may be standardized within a group of nodes including first node 602 and second node 604. The received idle information may include and/or reference an algorithm identifier. First node 602 and second node 604 may each maintain an association between one or more of the algorithm identifiers and a duration generator such as a function and/or a class configured to perform the identified algorithm.~~

~~{0063} A duration generator may determine the duration of time for detecting an idle time period based on one or more attributes accessible to one or both of first node 602 and second node 604. Exemplary attributes include a measure of network latency, a measure of network congestion, an indication of the availability of a particular resource, a user specified attribute, a security attribute, an energy usage attribute, a user attribute such as role of the user, and/or a measure of bandwidth supported by NIC 508 and/or a physical network medium operatively coupled to NIC 508.~~

~~[0064] Alternatively or additionally, idle information may include a parameter such as one or more of the attributes identified in the previous paragraph for use in a duration generator for determining a duration of time for measuring and/or otherwise detecting an idle time period.~~

~~[0065] A TCP connection may be identified by its endpoints. First node 602 and/or second node 604 may include an endpoint of the TCP connection. Alternatively or additionally, first node 602 and/or second node 604 may include a proxy endpoint representing an endpoint in a TCP connection. Nodes, that provide a network address translation (NAT) service, are exemplary nodes including proxy endpoints.~~

~~[0066] A node including a TCP connection endpoint is referred to as a host. Hosts are typically user devices and/or servers that typically operate at the edge of a network. While endpoints of most TCP connections are not typically included in network nodes for relaying, routing, and/or otherwise forwarding TCP packet data within a network such as routing nodes and switching nodes. Such network nodes may include one or~~

~~more connection endpoints for one or more respective TCP connections. It should be understood that the term "host" refers to a role played by a device in a network. First node 602 and/or second node 604 may play the role of a host in a TCP connection and/or may be proxy nodes.~~

~~[0067] A node is referred to as being in or included in a TCP connection when the node includes an endpoint of the connection and/or includes a proxy for a connection endpoint, referred to as a proxy endpoint. A proxy endpoint and an endpoint in a TCP connection may be in the same node or in different nodes.~~

~~[0068] In Fig. 5, connection state~~

3. One or more cost advisor components 402 operate in an execution environment 401. [0060] In Fig. 4a, cost advisor component 402a is illustrated as a component of application 403a. In Fig. 4b, cost advisor component 402b is illustrated as component of
24

network application agent 405b and/or browser 403b. In Fig. 4c, cost advisor

component 402c is illustrated operating external to one or more applications 403c. Execution environment 401c includes cost advisor component 402c in cost management subsystem 407c. In Fig. 4d, cost advisor component 402d is illustrated operating in network application 403d remote from a display device for presenting received information for updating a visual component. For example, cost advisor component 402d may operate in remote application provider node 506 while the received information is to be sent to a display device of user node 502 via network 504. [0061] Receiving resource information identifying a resource may include receiving an indication to present a representation of the resource to a user via an output device, receiving an indication identifying the resource as an input to a program component in an execution environment for performing an operation that includes processing the resource, detecting an access to the resource for the processing by a program component, detecting an input corresponding to a user

interface element including a representation of the resource, sending information to present a representation of the resource to a user via an output device, and/or intercepting a communication for accessing the resource. In response to one or more of these and/or analogous events, a cost advisor component 402 in Fig. 4a-d may receive resource information identifying a resource.

[0062] In an aspect, resource information identifying a second resource may be received. In response to receiving the resource information the first resource may be identified. The second resource may be identified as an alternative to processing the

25

first resource or as an additional resource to process along with the first resource based on a measure of a processing cost determined for one or both resources.

[0063] Figs. 4a-c illustrate that a cost advisor 402 may interoperate with a user interface component, such as a user interface element handler component 433. The user interface component may present a representation of a resource for selecting by a user as an input to a program component for performing an operation. For example, a selectable representation of a resource may be presented in an explorer or navigation window, a list box, a spinner, a text input box, a file selection dialog, and/or any other user interface component for selecting an item by a user. The presentation may be via

audio output with selection via a voice input device and/or other input device(s). Resource icons 608 in Fig 6 presented, based on received resource information, may illustrate selectable representations of one or more resources. Exemplary resources for processing by a program component include some or all of a data file, an executable file, a database record, a network message, input data, output data, a document, a media stream, a digital image, a communication communicated between at least two communicants, and/or a log.

[0064] In an aspect, a resource icon 608 in Fig. 6 may represent an image and/or a video to be processed by a program component, such as application 403a in Fig. 4a, for presenting as a background of a display region, such as a desktop or an application background. Cost advisor component 402a may present a file navigation user interface to receive resource information identifying one or more images and/or videos for processing by an IPU and/or other hardware component(s) included in execution

26

environment 401a as an input to application 403a for performing the operation of

presenting a desktop background.

[0065] In Fig. 4c, GUI subsystem 437c may receive resource information identifying an image and/or video to present in display presentation space

602 as desktop background 604. GUI subsystem 437c may interoperate with cost advisor component 402b to communicate resource information identifying a resource.

[0066] Network application agent 405b in Fig. 4b and/or network application 403d in Fig. 4d may receive resource information for presenting and/or otherwise processing by network application agent 405b and/or browser 403b. For example, cost advisor component 402b may be provided with resource information and/or cost advisor component 402d may be provided with resource information. Cost advisor component 402b and cost advisor component 402d may interoperate in an aspect. In another aspect, cost advisor component 558 may maintain state information for one or more TCP connection endpoints and/or proxy endpoints of corresponding TCP connections. 402b and cost advisor component 402d may operate independently. In still another aspect, one or the other of cost advisor component 402b and cost advisor component 402d may not be included in an instance adaptation of one or the other of execution environment, such as 401b and execution environment 502, 401d.

[0067] In various aspects and adaptations of cost advisor 302 in Fig. 3, such as cost advisors 402 in Figs. 4a-d, a cost advisor may be included in accessing and/or provided by first node 602 or second node 604.

~~[0069] First node 602 and/or second node 604 may play a role of a proxy node for a node including a TCP connection endpoint. First node 602 and/or second node 604 may include a proxy endpoint representing an endpoint in a TCP connection. A proxy node forwards TCP packet data, sent by a host including a TCP connection endpoint, to another host including a corresponding connection endpoint represented by a proxy endpoint included in the proxy node and vice versa. Exemplary proxy nodes in addition to including routing and/or switching capabilities may include a bridge, a hub, a repeater, a gateway, and a firewall.~~

~~[0070] In an aspect, a TCP keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and/or another timeout associated with a TCP connection may be modified based on the first idle information.~~

~~[0071] For example, in Fig. 5, ITP policy component 550 operating in first node 602 may modify an attribute of a TCP keep-alive option provided by one or more keep-alive components that may include settings service component 526. Modifying a keep-alive attribute may include creating the attribute, deleting the attribute, and/or modifying the attribute. ITP policy otherwise managing a resource. The cost advisor may receive resource information in response to an access to the resource. Resource information may be received through an invocation of a cost advisor 402 as a function, method, subroutine, and the like. The resource information may be received via a notification associated with a subscription to~~

27

events associated with the resource and/or a program component for processing the resource. The resource information may be received via an interprocess communication mechanism (IPC) such as a message queue, a pipe, a software

interrupt, and/or a hardware interrupt. The resource information may be received via a message received via a network.

[0068] Alternatively or additionally, resource information may be received in response to identifying a program component. A program component may process resources having a particular attribute, such as file type and/or content type. For example, application 403a in Fig. 4a may include a media player component to process files having file types and/or having content type identifiers that identify the media files including audio data, image data, and associated metadata. Resource information for one or more resources that match attribute information that identifies the resources as inputs for processing by a program component-550, may
~~interoperate with settings service component 526, connection state component 558, and/or a keep-alive option handler component (not shown) to detect the existence and state of one or more keep-alive attributes in determining whether a keep-alive option is active and/or in identifying its current state.~~

~~{0072} Inbe received by cost advisor component 402a in response to identifying the idle~~receiving ~~information, ITP policy component 550 may activate, disable, and/or modify the state of the keep-alive option via interoperation with one or more of settings service component 526, connection state component 558, and/or a keep-alive option handler. Thus, in response to identifying the idle information, ITP policymedia player ~~component 550 may prevent and/or alter the time a keep-alive packet is sent to second node 604 from first node 602.~~~~

~~{0073} Alternatively or additionally, ITP policy component 550 operating in first node 602 may modify an attribute associated with an acknowledgment timeout configured for TCP layer component 506. Modifying an acknowledgment timeout attribute may include creating the attribute, deleting the attribute, and/or modifying the attribute. ITP policy~~

~~component 550 may interoperate with settings service component 526, connection state component 558, and/or an acknowledgment option handler component (not shown) to detect the existence and state of one or more packet acknowledgment attributes. In response to identifying the idle information, ITP policy component 550 may modify the state of the packet acknowledgment option. Thus, in response to identifying the idle information, ITP policy component 550 may prevent and/or alter the time an acknowledgment is sent in a packet in a TCP connection(not shown).~~

[0069] A second resource may be identified based on the first resource. In one aspect, a program component may process resources having a particular attribute, such as file type and/or content type. For example, a drawing program component may process files having file types and/or content type identifiers that identify the files that are computer

drawn and/or editable via drawing. Resource information for one or more resources that match attribute information for an identified first resource may be received, in response to and/or otherwise based on identifying the first resource for processing and/or during

28

processing by a program component. For example, in Fig. 4c application 403c1 may be a graphics editing application. Application 403c1 may access a first resource having a file type and/or other content type identifier indicating that it includes editable graphics content. Cost advisor 402c may receive information identifying one or more other resources including editable graphics content in response to application 403c1 accessing the first resource.

[00740070] Returning to Fig. 2, block 204 illustrates ~~that~~ the method further includes ~~generating a TCP packet including a first idle time period header identifying metadata~~ determining at least one of a first measure of a specified processing cost for the processing of the first ~~idle time period based on the first idle information~~ resource and a second measure of the processing cost for the processing of the second resource. Accordingly, a system for ~~sharing information for detecting an idle TCP connection includes means for generating a TCP packet including a first idle time period header identifying metadata~~ selecting a resource based on a measure of a processing cost includes means for determining at least one of a first measure of a specified processing cost for the processing of the first ~~idle time period based on the first idle information~~ resource and a second measure of the processing cost for the processing of the second resource. For example, as illustrated in Fig. 4a, ~~packet generator component 4523, cost monitor component 304 is configured for generating a TCP packet including~~ determining at least one of a first ~~idle time period header identifying metadata~~ measure of a specified processing cost for the processing of the first ~~idle time period based on the first idle information~~ resource and a second measure of the processing cost for the processing of the second resource. Figs. 4a-d illustrate cost monitor components 404 as adaptations and/or analogs of cost monitor component 304 in Fig. 3. One or more cost monitor components 404 operate in an execution environment 401.

[0071] A metric defines a unit of measure. For example, an "inch" is a unit of measure for measuring length. A "kilowatt-hour" (kWh) is a unit of measurement in a metric for measuring an amount of energy. Instead of or in addition to measuring an amount a
29

metric may measure a rate. "Kilowatts per hour" (kWh/h) is energy or power metric for measuring a rate of energy used. A "measure" is a result of a particular measuring or measurement process. For example, 3 inches is a measure according to the length metric for inches, and 1000 kWh is a measure of an energy metric identifying an amount of energy. As used herein, a "measure of a processing cost" refers to a result of a measuring process for determining a processing cost according to a specified metric. Measuring may include estimating a measurement.

[0072] A processing cost may be determined and/or expressed by any metric, directly and/or indirectly providing an indication of a processing cost associated with processing a resource in performing a specified operation. A metric for determining a processing cost in terms of electrical power may be determined by monitoring a rate of electrical energy utilized over time period by a hardware component that is included in processing a resource. For example, a flow of electricity

to a network interface adapter may be monitored, for resources where processing the resources includes sending and/or receiving data via a network. Some of the data may be included in the resources. The

metric may represent the cost, for example, in kilowatt-hours, in kilowatts per hour, in transmission time, in bandwidth utilization, in latency, and/or in monetary units. In Figs. 4a-d, cost monitor component 404 may be invoked to determine and/or otherwise identify a measure of a processing cost for processing a particular resource in performing an operation.

[0073] A metric may be specified for measuring and/or expressing a processing cost in

a less direct manner. For example, with respect to energy cost, an energy cost may be

30

measured by counting occurrences of an energy consuming activity, such as a disk read. From another perspective a metric based on disk reads may be a direct measure of a utilization cost resulting from processing one or more resources stored in a hard- drive.

[0074] Exemplary metrics for measuring processing cost include metrics for energy, monetary metrics, time metrics, kinetic or stored energy metrics, heat metrics, metrics for resistance including mechanical and/or electrical resistance, metrics for measuring various energy and/or power consuming activities, metrics for measuring an environmental cost, health metrics, safety metrics, light metrics, metrics for measuring movement, metrics for measuring mass and/or weight, and/or metrics for measuring an opportunity cost.

[0075] A particular metric for determining a measure of a processing cost for a resource may be selected and/or otherwise identified based on one or more attributes of a resource, an operation that includes processing the resource, a program component for performing some or all of the operation, a hardware component included in processing the resource, a user, an organization, and/or a task; to name a few examples. For example, a metric such as a count of machine code instructions executed by an IPU may be specified and/or determined in performing a specified operation. An IPU based metric may be selected for measuring a cost of processing a resource where no output device is included in processing the resource. For an application or process that presents a user interface via a display component, a metric for measuring heat and/or light generated by the display device may be specified.

31

[0076] A cost monitor component 404, in an aspect, may determine a measure of a processing cost based on metadata provided in and/or with one or more of a resource, a program component for performing an operation that includes processing the resource, and a hardware component included in performing an operation that includes processing the resource. A measure of a processing cost may be predetermined and

located by cost monitor component 404 in and/or associated with a resource, a program component, and/or a hardware component. Cost monitor component 404 may access a table and/or other structure including predefined values for measures of the processing cost for a particular metric such as a change in temperature of a hardware component in Celsius and/or a time based cost represented in US dollars.

[0077] A cost monitor component 404 may look up and/or may otherwise identify a predefined value based on a type of a resource, a size of a resource, an energy source, a hardware component, and/or a program component for processing the resource. The predefined value may be a measure of a processing cost and/or may be an input for determining a measure of a processing cost expressed according to an identified metric. For example, a predefined value may be multiplied by a measure of time that a resource may be processed by a program component to produce a time based metric such as disk reads per minute.

[0078] In another aspect, a cost monitor component 404 may determine a measure of a processing cost by calculating the measure according to the specified metric and/or may interoperate with a sensor, such as a thermometer, in measuring a cost of processing. Cost monitor component 404 may include and/or otherwise access one or

32

more measurement components for determining a measure according to one or more metrics.

[0079] In Fig. 4a, cost monitor component 404a is illustrated operating in application 403a. Cost monitor component 404a may determine a measure of a processing cost for resources processed by application 403a. The resources may include resources provided by application 403a to other applications, subsystems, and/or components operating in execution environment 401a and/or in another execution environment

~~{0075} Fig. 5 illustrates packet generator component 552 as an adaptation of and/or analog of packet generator component 452 in Fig. 4a. One or more packet generator components 552 operate in execution environment 502.~~

~~{0076} Packet generator component 552 in Fig. 5 may receive idle information and/or information based on the received idle information from ITP policy component 550.~~

~~Whether and when packet generator component 552 receives information for including an idle time period (ITP) header in a TCP packet may depend on a current state of the associated TCP connection. In Fig. 5, ITP policy component 550 may interoperate with connection state component 558 to determine whether and when to provide information to packet generator component 552 for including an ITP header in a TCP packet.~~

~~{0077} In an aspect, an ITP header may be included in a packet exchanged during setup of a TCP connection. RFC 793 describes a "three way handshake" for establishing a TCP connection. Other message exchanges may be used in setting up a TCP connection as those skilled in the art will understand. Such other exchanges are not currently supported by the TCP as described in RFC 793. The~~

~~specified "three-way handshake" and other patterns of message exchange for setting up a TCP connection include packets that are considered to be in the TCP connection for purposes of this disclosure. Including an ITP header may be restricted to packets exchanged in connection setup, excluded from packets exchanged during connection establishment, or allowed in one or more packets exchanged during connection establishments and in packets exchanged after connection setup.~~

~~[0078] In an aspect, when connection state component 558 and/or ITP policy component 550 determine an ITP header should be included in a TCP packet based on received idle information, packet generator component 552 may include the ITP header in a next TCP packet generated in response to data received via application in port component 522 for sending to first node 602. In another aspect, packet generator~~

~~component 552 may send the ITP header in a TCP packet in the TCP connection with no data included in the TCP data stream sent by first node 602 to second node 604. Such a packet is referred to as an empty packet. Packet generator component 554 may send the empty TCP packet when TCP layer component 506 has no fer data from an application~~
included in and/or otherwise provided by one or more devices.

[0080] For example, application 403a may present selection window 606 in Fig. 6. The resources processed by application 403a may include data, represented by resource icons 608, to send to another node. Cost monitor component 404a may determine a measure of a processing cost for transmitting the data to the other node via a particular physical network media physically coupled to a network adapter in execution environment 401a. Cost monitor component 404a may be configured with and/or otherwise may determine a measure based on a count of bytes in the resources and/or in an encoded translation of the resource(s) for transmitting.

[0081] In Fig. 4b, cost monitor component 404b is illustrated operating at least partially in network application agent 405b. Some or all of network application agent 405b may be received by browser 403b, operating in seconduser node 604 to send in the TCP data stream to first node 602.

~~[0079] Packet generator component 552 may generate a packet according to the TCP specifications and may include a header identified as an ITP header in accordance with specifications for including TCP option headers in a TCP packet. See RFC 793 for more details. Fig. 8 illustrates a format or structure for a TCP packet 802 as described in RFC 793. Each "+" character in Fig. 8, indicates a bit boundary. TCP packet 802 specifies a location and format for including a source port 804 portion including an identifier for an endpoint of the TCP connection for a sending node and a destination port 806 including an identifier for a corresponding endpoint of the TCP connection in a receiving node. IP packet 810 illustrates a format for an IP packet header for an IP packet including TCP packet data. Source address 812 specifies a location and format in an IP header for including a network address identifying a network interface of the sending node, and destination address 814 identifying a network interface for the receiving node. A network address and a port number identify a connection endpoint in a network. Two endpoints identify a TCP connection.~~

~~[0080] Fig. 8 also illustrates a format for an exemplary ITP header 820. A KIND field 822 is specified for including an identifier indicating that the option is an idle time period~~

~~(ITP) option in an ITP header. Identifiers for option headers are currently under the control of the Internet Assigned Numbers Authority (IANA). Length field 824 identifies a length of an ITP header. An ITP data field 826 is specified for including ITP header information for detecting an idle time period as described herein.~~

~~[0081] Those skilled in the art will recognize given this disclosure that an ITP header may have other suitable formats and may be included in a TCP packet in structures and locations other than those specified for TCP options in RFC 793. An equivalent or analog of an ITP header may be included in a footer of a protocol packet in an extension and/or variant of the current TCP.~~

~~[0082] ITP data field 826 in Fig. 8502 in Fig. 5, from network application 403d in Fig. 4d operating in application provider node 506 in Fig. 5 as described above. Fig. 4d illustrates cost monitor component 404d operating in network application 403d. Cost monitor component 404b and/or cost monitor component 404d~~

~~33~~

may determine a measure of a processing cost for resources processed by network application 403d and/or network application agent 405b. Cost monitor component 404b and/or cost monitor component 404d may be components in a cost monitor system distributed between network application agent 405b and network application 403d. Cost monitor component 404b and/or cost monitor component 404d may operate independently. Operating independently may include one of the cost monitor components operating in the absence of the other cost monitor component. Resources may include resources provided by network application agent 405b to browser 403b and/or extensions of browser 403b. The resources may further include resources provided to other nodes in network 504 by network application agent 405b and/or network application 403d. Network application agent 405b may interoperate with browser 403b to present selection window 606 in Fig. 6 in a browser window or tab (not shown). The resources processed may be represented by resource icons 608.

[0082] Cost monitor component 404b and/or cost monitor component 404d may determine a measure of a processing cost for processing one or more resources where processing the resources includes transmitting the resources via network 504 in Fig. 5. Transmitting resource data may include encoding, decoding, filtering, translating, and/or transforming some or all of the data in a resource in some manner. For example, a resource may be compressed prior to transmitting via network 504. Cost monitor component 404b may determine a metric based on a type of physical layer network included in network 504, may determine a metric based on an encoding, decoding, and/or other transformation, may determine a metric based on a manufacturer and/or

34

type of network interface component, and/or may determine a metric based on network throughput data and/or other network attributes and/or metadata. The measure may be a cost for transmitting a web document via a network including a modem, a cost for retrieving image data in the web document from a hard drive, a cost for decoding data received via network 504, and/or a cost for transmitting data over a secure network connection. Cost monitor component 404b and/or cost monitor component 404d may be configured with and/or otherwise may identify a predefined measure of a processing cost according to a metric selected by a developer of browser 403b and/or based on version information for browser 403b.

[0083] In Fig. 4c, cost monitor component 404c is illustrated operating in cost management subsystem 407c. Cost management subsystem 407c may be a subsystem of execution environment 401c that provides services to a number of program components operating in execution environment 401c and/or in another execution environment communicatively coupled to a network 504 in Fig. S. Cost monitor component 404c may determine measures of a processing cost(s) for various resources processed by various applications 403c operating in execution environment

401c. The resources may include resources provided to and/or otherwise
~~identify metadata for the first idle time period. For example, an ITP data field in a packet may include and/or otherwise identify one or more of a duration of time for detecting an idle time period, a duration generator for determining a duration of time for detecting an idle time period, and a parameter for use in a duration generator for determining a duration of time for measuring and detecting an idle time period.~~

~~[0083] Message 704 in Fig. 7 illustrates an invocation and/or other access to packet generator component 552 for generating a TCP packet including an ITP header based on received idle information~~accessible to applications 403c via various subsystems of execution environment 401c, such as a file system (not shown) and/or network stack 411c.

[0084] For example, selection window 606 in Fig. 6 may be presented as a document navigation window presented by execution environment 401c. "Op1" presented in operation bar 610 may invoke first application 403c1 for processing a currently selected

35

resource, illustrated as resource icon 6082b. "Op2" in operation bar 610 may be a user interface control for invoking second application 403c2 for one or more selected resources represented by resource icons 608. Resources processed by first application 403c1 may include documents having various content types. Cost monitor component 404c may determine a measure of a processing cost for the documents expressed by a metric based on the format of the respective documents identified by a content type and based on an operation for processing the documents. The operation may be performed by application 403c1 and/or may be performed by one or more other components. [0085] For example, for a particular device a file system operation may be configured to be an operation for determining a measure of a processing cost for a resource processed by first application 403c1. In Fig. 4c, cost

management subsystem 407c may determine processing costs for resources, freeing applications 403c from determining processing costs. Note that, in an aspect, at least some of cost management subsystem 407c may operate in a node other than the node included in and/or providing execution environment 401c. For example, some or all of the arrangement of components may be adapted to operate in execution environment 401d, which includes and/or is otherwise provided by application provider node 506.

[0086] Cost monitor component 404a is illustrated operating in application 403a. Cost monitor component 404a may determine a measure of a processing cost for resources processed by application 403a. Application 403a is a program component and may include one or more program components. The resources may include resources provided by application 403a to other applications, subsystems, and/or components

36

operating in execution environment 401a and/or in another execution environment included in and/or otherwise provided by one or more devices. For example, application 403a may present selection window 606 in Fig. 6. The resources processed by application 403a may include images and videos represented by background resource icons 608. Cost monitor component 404a may determine a measure of a processing cost for presenting the images and/or videos as desktop backgrounds. For example, cost monitor component 404a may be configured with and/or otherwise may determine a measure based on a count of display refreshes over a specified period of time for the various resources to determine a measure of a processing cost for presenting the various respective resources.

[00840087] Returning to Fig. 2, block 206 illustrates ~~that~~ the method ~~yet~~ further includes ~~-sending the TCP packet in the TCP connection to the second node to provide the metadata for~~ selecting one of the first resource and the second resource based on the at least one of the first ~~idle time period to measure and~~ the second ~~node measure~~. Accordingly, a system for ~~sharing information for detecting an idle TCP connection further~~ selecting a resource based on a measure of a processing cost includes means for ~~sending~~ selecting one of the ~~TCP packet in the TCP connection to~~ first resource and the second node to provide the metadata for ~~resource based on the at least one of~~ the first ~~idle time period to measure and~~ the second ~~node measure~~. For example, as illustrated in Fig. 4a3, the ~~net out port~~ cost director component 454306 is configured for ~~sending~~ selecting one of the ~~TCP packet in the TCP connection to~~ first resource and the second ~~node to provide the metadata for~~ resource based on the at least one of the first ~~idle time period to measure and~~ the second ~~node measure~~. Figs. 4a-d illustrate cost director components 406 as adaptations and/or analogs of cost

director component 306 in Fig. 3. One or more cost director components 406 operate in

~~{0085} Fig. 5 illustrates net out port component 554 as an adaptation of and/or analog of net out port component 454 in Fig. 4a. One or more net out port components 554 operate in execution environment 502. Net out port component 554 is illustrated operatively coupled to packet generator component 552. Net out-port component 554 may receive TCP packet data from packet generator~~

~~component 552 and interoperate with IP layer component 514 to send the TCP packet in one or more IP packets via network 606 to second node 604. Message 706.1 in Fig. 7 illustrates a TCP packet including an ITP header sent by first node 602 and received by second node 604.~~

~~{0086} In one aspect, an ITP header may be sent to make sending one or more TCP keep-alive packets by a partner node in the connection unnecessary. A receiver of a packet including an ITP header, such as second node 604, may keep a TCP connection alive based on information in the ITP header~~

~~{0087} In another aspect, first node 602 may set a keep alive timeout attribute based on a duration of the first idle time period identified in the first idle information and/or in~~

~~the metadata provided to second node 604. For example, first node 602 may monitor a time period during which no non-empty packets are sent or received in the TCP connection. A keep-alive option handler and/or keep-alive component (not shown) operating in first node 602 may set a keep-alive timer according to the timeout attribute, with a duration that will result in the keep-alive timer expiring before an idle time period can occur. In response to detecting a keep-alive timeout, which may be indicated by the expiration of the keep-alive timer, the keep-alive option handler and/or keep-alive policy component may provide information to packet generator component 552 to generate a TCP keep-alive packet. The packet generator component 552 may provide the generated packet to net out port component 554 for sending the TCP keep-alive packet to second node 604 to determine whether the TCP connection is active and/or to keep the TCP connection active.~~

~~{0088} In another aspect, ITP policy component 550 operating in first node 602 may set a timer, analogous to the keep-alive timer described in the previous paragraph that expires before an idle time period can occur. In response the timer expiring, ITP policy component 550 may provide idle information to packet generator component 552 to generate a TCP packet including a second ITP header. Content of the second ITP header may be based on the first idle information received, data received from second node 604, information received from a network application that may be from a user, and/or on any information accessible to TCP layer component 506 in execution environment 502 in first node 602. The TCP packet generated by packet generator~~

~~component 552 is provided to IP layer component 514 via net out port component 554 to send to second node 604 in the TCP connection. Along with sending the message, first node 602 may reset and/or otherwise restart detection of the first idle time period. Thus, a second ITP header may be sent in a second TCP packet by first node 602 to second node 602 along with restarting detection of the first idle time period. Alternatively, first node 602 may reset and initiate detection of an idle time period with a different duration than the previous idle time period, based on the idle information for generating the second ITP header.~~

~~{0089} Returning to Fig. 2, block 208 illustrates the method further includes detecting the first idle time period based on the first idle information. Accordingly, a system for sharing information for detecting an idle TCP connection further includes means for detecting the first idle time period based on the first idle information. For example, as illustrated in Fig. 4a,~~

~~the idle time period monitor component 456 is configured for idle time period monitor.~~

~~{0090} Fig. 5 illustrates idle time period monitor component 556 as an adaptation of and/or analog of idle time period monitor component 456 in Fig. 4a. One or more idle time period monitor components 556 operate in execution environment 502.~~

~~{0091} In an aspect, in response to receiving the first idle information, ITP policy component 550 may store a value representing a duration of time in a configuration storage location. Alternatively, or additionally, ITP policy component 550 may invoke a duration generator to determine a duration of time for detecting the idle time period. The~~

~~duration generator may be preconfigured for the TCP connection and/or may be identified based on the idle information received. As described, the invoked generator may be invoked with a parameter included in and/or otherwise identified based on the received idle information.~~

~~{0092} ITP policy component 550 may interoperate with ITP monitor component 556 to identify the duration for detecting the idle time period. ITP monitor component 556, in various aspects, may receive information including and/or otherwise identifying a duration of time, a duration generator, and/or a parameter for a duration generator. ITP monitor component 556 may initiate and/or restart a process for detecting an idle time period. In an aspect, ITP monitor component 556 detects and/or otherwise identifies a beginning of a potential idle time period based on one or more specified events.~~

~~{0093} In an aspect, detecting the first idle time period by ITP monitor component 556 may include detecting a time period in the idle time period during which first node 602 has received acknowledgment for all data sent via the TCP connection in the TCP data stream by first node 602 to second node 604. Further, the first idle time period may include a time period during which first node 602 has sent one or more TCP packets to second node 604 to acknowledge all data received in a TCP data stream in the TCP connection from second node 604 to first node 602. Detecting the first idle time period by ITP monitor component 556 may include detecting that all received data has been acknowledged and/or that all sent data has been acknowledged.~~

~~{0094} In an aspect, ITP policy component 550 may include a policy with a rule indicating that an idle time period cannot begin while a TCP packet sent by first node 602 remains unacknowledged by second node 604. ITP policy component 550 may prevent ITP monitor component 556 from initiating detection of an idle time period while unacknowledged data exists. In a further aspect, a time duration may be associated and/or included in the policy identifying a limit to a period of waiting to receive acknowledgment of TCP packet data sent by first node 602. In one aspect, waiting for lack of an acknowledgment for an empty packet does not delay detection of an idle time period, while in another aspect ITP monitor component 556 will not initiate detection while an empty packet remains unacknowledged.~~

~~{0095} In an aspect, idle information, received by a node may be included in and/or otherwise based on a previous idle time period header identified in a previous TCP packet received in the TCP connection by the node from a remote~~

~~node prior to sending an ITP header based on the idle information by the node. For example, the first idle information received by ITP policy component 550 in first node 602 may be based on an idle time period header included a TCP packet in the TCP connection sent by second node 604 and received by first node 602 prior to sending the first TCP packet by first node 602. The exchange of ITP headers may include a negotiation between first node 602 and second node 604.~~

~~{0096} A duration of time may be identified based on the idle information received by ITP policy component in first node 602. A timer may be set according to the identified~~

~~duration. Detecting the first idle time period may include and/or otherwise may be based on detecting the timer expiration. ITP monitor component 556 may set a timer configured to expire in a time duration identified based on the first idle information received by ITP policy component 550. The identified duration may be longer, shorter, or equal to a duration of the idle time period. ITP monitor component 556 may use multiple timers. ITP monitor component 556 may recalculate and/or otherwise generate a new idle duration based on the idle information at one or more times during detection of the first idle time period. That is, a duration of an idle time period may be static and/or may be dynamic, changing based on attribute information accessible during the detection process and/or based on one or more duration generators.~~

~~{0097} Message 710.1 illustrates a call and/or other communication between ITP monitor component 556 and a timer component in first node 602 to set a timer included in detecting an idle time period. Prior to the setting the timer, first node 602 and second node 602 may be active in exchanging TCP packets as illustrated by messages including message 706.2 through message 706.n. Those skilled in the art will recognize that detection of an idle time period may not include explicitly and/or directly using a timer. ITP monitor component 556 may monitor other events as a proxy or indirect mechanism for initiating detection and detecting an idle time period.~~

~~{0098} ITP monitor component 556 may detect one or more events configured to indicate that an idle time period has occurred. For example, expiration of a timer or multiple associated timers may be interpreted by ITP monitor component 556 as~~

~~marking an occurrence of the first idle time period. Message 710.2 illustrates ITP monitor component 556 receiving information identifying expiration of a timer for detecting the first idle time period.~~

~~{0099} In a further aspect, in response to detecting the expiration of a timer set as described above, a TCP keep-alive packet may be sent by first node 602 to determine whether the TCP connection is active and/or to keep the TCP connection active. When the keep-alive packet is sent, an acknowledgment timer may be set. If a timeout of the acknowledgment timer is detected indicating no TCP packet has been received acknowledging the keep-alive packet, the first idle time period may be detected in response to and/or otherwise based on the timeout of the acknowledgment timer.~~

~~{0100} In Fig. 5, ITP policy component 550 in first node 602 may provide a duration identified based on the received idle information to a keep-alive~~

~~monitor component (not shown). The keep-alive monitor component may configure a keep-alive timer to expire based on the identified duration. In response to detecting expiration of the keep-alive timer, ITP monitor component 556 may invoke packet generator component 552 to generate a TCP keep-alive packet. First node 602 may send the TCP packet to second node 604. The TCP keep-alive packet may be sent to prevent detection of an idle time period by second node 604 and/or may otherwise be sent to detect by first node 602 whether the TCP connection is active.~~

~~{0101} First node 602 may set an acknowledgment timer associated with sending the packet. If the acknowledgment timer expires before a TCP packet is received from~~

~~second node 602 acknowledging the packet sent, ITP monitor component 556 may detect the idle time period in response to and/or otherwise based on expiration of the acknowledgment timer.~~

~~{0102} Receiving a packet from second node 604 included in the TCP connection is an event that, in various aspects, may directly and/or indirectly indicate the beginning of a potential idle time period. A potential idle time period may begin at some specified point during and/or after processing a received TCP packet. In one aspect, an empty TCP packet may be received while a potential idle time period is being monitored. That is, a beginning of the potential idle time period has been detected. In response to receiving the empty TCP packet, monitoring of the current potential time period may be aborted. Further, in response to receiving the empty TCP packet, a beginning of a next potential idle time period may be detected.~~

~~{0103} In Fig. 5, ITP policy component 550 and ITP monitor component 556 may operate to reset and/or initiate detection of an idle time period in response to receiving an empty TCP packet. First node 602 may receive an empty packet. In response, ITP monitor component 556 may receive an event and/or other indication to reset detection of an idle time period. Resetting the detecting process may be based on whether or not a received empty TCP packet matches a specified condition. ITP option handler component 562 may be configured to determine whether a received empty TCP packet matches the condition. If ITP option handler component 562 determines the empty packet matches the condition, ITP monitor component 556 may be instructed to reset~~

~~and/or restart detection of the first idle time period including detecting the beginning of a next potential time period.~~

~~{0104} The condition may match received TCP packets including ITP headers and/or other TCP option headers. A condition may match a port number and/or other field in TCP packet. A condition may further be based on a network address in an IP header including the TCP packet.~~

~~{0105} In a further aspect, first node 602 may receive via network 606 from second node 604 a TCP packet in the TCP connection including an second ITP header. Message 706.2 in Fig. 7 illustrates the TCP packet sent by second node 604. ITP option handler component 562 may identify the second ITP header received from second node 604. The identified second ITP header may be for detecting by first node 602 an idle time period, during which no TCP packet in the TCP connection is received, by the first node 602 that includes data~~

~~in the first TCP data stream from second node 604. The first idle time period may be detected by ITP monitor component 556 in first node 602 based on the second ITP header and based on the received idle information. The second ITP header received in the TCP packet from second node 604 may be based on the first ITP header in the TCP packet sent in the TCP connection by first node 602 to second node 604.~~

~~[0106] In some aspects, the first node and second node 604 may continue to exchange ITP headers. Information in the exchanged ITP headers may be based on ITP headers received in the TCP connection and/or on data accessible locally to one or~~

~~both of the nodes. In some aspects, the exchange may be a negotiation while in other the exchange may simply be informational.~~

~~[0107]~~

an execution environment 401.

37

[0088] In an aspect, a resource may be selected and/or otherwise identified based on selection information received in response to a user input detected by an input device and based on a measure of a processing cost. The measure and/or an indication based on the measure may be presented for respective resources for processing by a particular program component. For example, a user may open and/or otherwise initiate operation of an application 403a in Fig. 4a. Application 403a may present, in selection window 606 in Fig. 6, a user interface for selecting one or more resources from multiple resources for processing by the application. For example, cost director component 406a in Fig. 4a may send presentation information to present a cost indication for the one or more resources presented in selection window 606 in Fig. 6, allowing a user to select a resource represented by a particular resource icon 608 from the resources represented by respective resource icons. Selection input from a user identifying a resource represented by the particular icon may be received and/or otherwise selected based on a cost indication presented for the resource. The cost indication is based on a measure

of a processing cost for the resource.

[0089] A user input selecting a resource for processing by the application may be detected by a UI element handler 433a for the selection user interface and/or for the representation of the resource. In an aspect, cost director component 406a may determine whether to send presentation information to present a cost indication for a resource based on a cost condition, such as an energy condition. For example, cost monitor component 404a may evaluate a specified energy condition based on a determined measure of a processing cost for a resource. When the energy condition is

met, cost director component 406a interoperating with cost monitor component 404a may select one or more resources for processing and send presentation information for presenting the resource in the selection user interface. When the condition is not met, the resource is not selected for presenting. Thus a selectable representation of a resource may be a cost indication based on a measure of a cost of processing. In the aspect, only resources that meet a particular energy condition may be selected. For example, in Fig. 6 normally presented resource icons 608 may be presented as selectable as directed by cost director component 406. Pattern icons such as icon 608mb may be presented as non-selectable by cost director component 406, so that user input corresponding to icon 608mb is not processed as a selection of the resource as analogous input for normally presented icons is processed as a selection. [0090] Fig. 6 also illustrates presenting a cost indication based on a measure of a processing cost for a resource. The measure may be determined based on a metric for measuring a particular cost of processing and presented as a numeric indication based on the measure. Cost indications determined for the respective resources are illustrated by resource icons 608 in Fig. 6. For example, the illustrated indications in Fig. 6 are based on a five-point scale providing relative indications of a processing cost for the respective resources. An indication of "5" may be defined as a cost indication for a most expensive resource or resources for processing according the metric. A "1" indication may indicate resources that require the least cost according to the metric as presented according to the five-point scale. The cost indications may change based on an operation selected in operation bar 610. A resource may be selected for presentation in

a region based on a cost indication. A user may select a resource for processing from the presented resource representations based on the presented cost indications.

[0091] A region of display presentation space 602 may be designated for presenting a resource having a cost that matches a specified cost condition. For example, higher- cost resources may be placed relatively closer to the bottom of a screen than relatively lower-cost resources. A position in a screen may be a cost indication. In another aspect, an orientation of a UI element representing and/or associated with a resource may be defined as a cost indication based on a specified measure according to a particular metric. Variations in other user detectable attributes may be configured as indications for various metrics in other aspects.

[0092] A cost director component 406 may change or otherwise provide for changing a pointing device representation, such as a mouse pointer, when it approaches and/or is in a location of a presented resource, as a cost indication based on a measure according to a particular metric for processing a resource represented by a UI element in the location. For example, different colors of the pointer may be defined as different indications associated with different measures of a processing cost. Alternatively or additionally, a

pointer may be deactivated for selecting a resource based on a measure of a processing cost for the resource. In an aspect, a resource may be automatically selected when a cost condition based on a measure of a processing cost is met. Thus automatic selection may be a cost indication.

[0093] In Fig. 4d, cost director component 406d in network application 403d may send information via a response to a request and/or via an asynchronous message to a

40

client, such as browser 403b and/or network application agent 405b, to present a user detectable indication of a measure of a processing cost for a resource. One or more resources may be selected for representing. The one or more resources may be selected based on their respective measures and/or corresponding cost indications.

[0094] In another aspect, a user input for selecting a resource may be detected. A warning indication may be presented, in response to receiving the selection, when an energy condition, based on a measure of energy for the resource, is not met. A cost director component 406 may direct a UI element handler component 433 to present a warning when cost monitor component 404 determines that an energy condition is not met for the selected resource, based on a determined measure of a processing cost for the resource. The measure of a processing cost may be based on an energy metric. [0095] An indication of a measure of a processing cost for a resource may include presenting a representation of the resource in a plurality of representations of resources according to an order of respective measures of processing costs determined for the resources in the plurality.

[0096] Presentation information for presenting a cost indication may be sent in a message via a network to a node operatively coupled to an output device. Cost director component 406d in Fig. 4d operating in application provider node 506 in Fig. 5 may send presentation information in response to a request from network application agent 405b in Fig. 4b operating in user node 502.

[0097] A change in a measure of a processing cost associated with a resource being processed and/or change in a measure of a processing cost of an alternative and./or

41

additional resource may be detected. In response to one or more detected changes, an alternative and/or an additional resource may be selected for respectively identifying to a program component instead of and/or in addition to the resource currently being processed.

[0098] For example, cost management subsystem 407c in Fig. 4c may monitor a level of energy in a battery providing energy for a handheld device. Cost management subsystem 407c may invoke cost monitor component 404c in response to detecting a battery energy level falling below a specified threshold. Cost monitor component 404c may recalculate and/or otherwise determine a measure of a processing cost for one or more resources in response to the change in

battery state. Cost monitor component 404c may provide changed measures of a processing cost to cost director component 406c. Cost director component 406c may select an alternative and/or an additional resource, depending on the change to cost operations component 408c, for identifying to the program component.

[0099] In an aspect, an energy condition may be specified. A measure of a processing cost determined for a resource may be determined for evaluating an energy condition to determine whether the energy condition is met. An energy condition may be identified for evaluating and/or may be evaluated based on an energy source, an amount of energy available, an amount of energy available in a battery and/or other energy store, an amount of energy used and/or currently being used for processing another resource, a location of the device, and a time required for restoring an energy store to a specified

42

state, to name a few examples. A location of a device may be a location with respect to

another location for charging or changing energy sources.

[0100] When an energy condition and/or other cost condition is not met for a resource, a cost director component 406 may select one or more alternative resources to a first resource based on one or more respective measures of a processing cost determined by a corresponding cost monitor component 404. One or more representations of the respective one or more alternative resources may be selected for presenting to a user. A user input may be received from the user for selecting an alternative resource. A user input may be received indicating that the first resource is to be provided to the program component for processing. In yet another alternative, a cost director component 406 may automatically select an alternative resource that meets the energy condition and/or

other cost condition for processing instead of the first resource. In an additional aspect, the first resource may meet the energy condition and cost director component 406 may select one or more resources from the alternatives to identify to the program component in addition to the first resource. Cost director component 406 may identify the additional resource(s) automatically or may receive input from a user to identify one or more additional resources. [0101] Also as described above, a resource may be selected automatically, based on a measure of a processing cost for the resource, by a cost director component 406. Cost operations component 408 may access the resource, transform the resource into a format suitable for processing by the program component, and/or otherwise identify the resource to the program component for processing.

43

[0102] Returning to Fig. 2, block 210208 illustrates that the method yet further includes ~~deactivating the TCP connection in response to detecting~~

~~the first idle time period~~identifying, to the program component, the selected one of the first resource and the second resource for processing. Accordingly, a system for ~~sharing information for detecting an idle TCP connection further includes means for deactivating the TCP connection in response to detecting the first idle time period~~selecting a resource based on a measure of a processing cost includes means for identifying, to the program component, the selected one of the first resource and the second resource for processing. For example, as illustrated in Fig. 4a³, ~~the connection state~~cost operations component 458³⁰⁸ ~~is configured for deactivating the TCP connection in response to detecting the first idle time period.~~

~~{0108} Fig. 5 illustrates connection state component 558 as an adaptation of and/or analog of connection state component 458 in Fig. 4a. One or more connection state components 558 operate in execution environment 502.~~

~~{0109} When ITP monitor component 556 in first node 602 detects an idle time period, ITP monitor component 556 may provide an indication to connection state component 558. The indication may indicate that the idle time period for the TCP connection has been detected and/or otherwise may instruct connection state component 558 and/or other components in TCP layer component 506 to deactivate the TCP connection. Message 712 in Fig. 7 illustrates a communication to deactivate the TCP connection communicated in response to detecting the idle time period.~~

~~{0110} Deactivating the TCP connection may include closing the TCP connection. A TCP connection may be closed using a three-way handshake packet exchange described in RFC 793. Deactivating the TCP connection may include sending a TCP packet by the detecting node to reset the TCP connection. According to RFC 793, first node 602 may send a TCP packet including a reset (RST) bit set to "1" to indicate a connection reset. Deactivating the TCP connection may include, alternatively or additionally, releasing a resource allocated for maintaining and/or activating the TCP connection.~~

~~{0111} With respect to the method illustrated in Fig. 3, block 302 illustrates the method includes receiving, by a second node from a first node, a first transmission control protocol (TCP) packet in a TCP connection. Accordingly, a system for sharing information for detecting an idle TCP connection includes means for receiving, by a second node from a first node, a first transmission control protocol (TCP) packet in a TCP connection. For example, as illustrated in Fig. 4b, the net in port component 460 is configured for receiving, by a second node from a first node, a first transmission control protocol (TCP) packet in a TCP connection.~~

~~{0112} Fig. 5 illustrates net in port component 560 as an adaptation of and/or analog of net in port component 460 in Fig. 4b. One or more net in port components 560 operate in execution environment 502.~~

~~{0113} As described above, net in port component 560 in Fig. 5 may operate in an instance of execution environment 502 and/or an analog included in and/or including~~

~~second node 604. The TCP packet, illustrated by message 706.1 in Fig. 7 and described above, may be received by net in port component 560 in second node~~

~~604. The TCP packet may include data in a second TCP data stream sent by first node 602 to second node 604 to deliver to a user of TCP layer component 506 in second node 604 such as network application 504. Alternatively, the TCP packet may be an empty TCP packet. The received TCP packet may be a packet included in setting up the TCP connection as described above.~~

~~{0114} Returning to Fig. 3, block 304 illustrates the method further includes detecting a first idle time period header, in the first packet, identifying metadata for a first idle time period, detectable by the first node, during which no TCP packet including data in a first TCP data stream sent in the TCP connection by the second node is received by the first node. Accordingly, a system for sharing information for detecting an idle TCP connection includes means for detecting a first idle time period header, in the first packet, identifying metadata for a first idle time period, detectable by the first node, during which no TCP packet including data in a first TCP data stream sent in the TCP connection by the second node is received by the first node. For example, as illustrated in Fig. 4b, idle time period option handler component 462 is configured for detecting a first idle time period header, in the first packet, identifying metadata for a first idle time period, detectable by the first node, during which no TCP packet including data in a first TCP data stream sent in the TCP connection by the second node is received by the first node.~~

~~{0115} Fig. 5 illustrates idle time period option handler component 562 as an adaptation of and/or analog of idle time period option handler component 462 in Fig. 4b. One or more idle time period option handler components 562 operate in execution environment 502.~~

~~{0116} In Fig. 5, ITP option handler component 562 is operatively coupled to packet handler component 516. The TCP packet, including the ITP header sent by first node 602, may be received, and identified as a TCP packet by net in port component 560 operating in second node 604. As illustrated in Fig. 5, net in port component 560 and/or an analog of net in port component 560 may provide and/or otherwise identify the received packet to packet handler component 516. Packet handler component 516 may detect various portions of the TCP packet according to the TCP packet 802 structure as illustrated in Fig. 8. Alternatively, packet handler component 516 may provide some or all of the packet to various components in TCP layer component 506 to identify portions of the packet according to the TCP specification and/or according to a particular implementation.~~

~~{0117} The ITP header sent by first node 602 may be received by and/or otherwise identified by ITP option handler component 562. Message 708 in Fig. 7 exemplifies activation of ITP option handler component 562 for detecting the ITP header in the TCP packet received from first node 602 by second node 604.~~

~~{0118} In various aspects, ITP option handler component 562 operating in second node 604 may detect and/or otherwise determine a duration of time associated with~~

~~detection of the idle time period by first node 602, a duration generator, and/or a parameter for a duration generator. The first idle time period header may identify metadata including and/or identifying, for detection of the first idle time period by first node 602, a duration of time, a generator for~~

~~determining a duration of time, and/or an input for determining a duration of time.~~

~~[0119] Returning to Fig. 3, block 306 illustrates the method yet further includes modifying, based on the metadata, by the second node a timeout attribute associated with the TCP connection. Accordingly, a system for sharing information for detecting an idle TCP connection includes means for modifying, based on the metadata, by the second node a timeout attribute associated with the TCP connection. For example, as illustrated in Fig. 4b, the option attribute handler component 464 is configured for modifying, based on the metadata, by the second node a timeout attribute associated with the TCP connection.~~

~~[0120] Fig. 5 illustrates option attribute handler component 564 as an adaptation of and/or analog of option attribute handler component 464 in Fig. 4b. One or more option attribute handler components 564 operate in execution environment 502.~~

~~[0121] In an aspect, ITP option handler component 562 may one or more attribute option handler components 564 to modify one or more corresponding attributes of a keep alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and another timeout associated with the TCP connection, in response to~~

~~identifying the ITP header. The modifying may be based on the content of the ITP header.~~

~~[0122] For example, ITP option handler component 562 in second node 604 may interoperate with a keep alive attribute option handler component 564 directly and/or indirectly via settings service component 526, connection state component 558, and/or a keep alive policy component (not shown) to detect the existence and state of one or more keep alive attributes in determining whether the keep alive option is active and/or the state of the keep alive option.~~

~~[0123] In response to identifying the idle time period header, ITP option handler component 562 may activate, disable, and/or modify the state of the keep alive option via interoperation with the keep alive attribute option handler. Thus, in response to identifying the idle information, attribute option handler component 564 may prevent and/or alter the time a keep alive packet is sent by second node 604 to first node 602.~~

~~[0124] Alternatively or additionally, an attribute option handler component 564 may modify an attribute associated with a packet acknowledgment option provided by TCP layer component 506 in first node 602. Modifying a packet acknowledgment attribute may include creating the attribute, deleting the attribute, and/or modifying the attribute. Attribute option handler component 564 may interoperate with settings service component 526, connection state component 558, and/or an acknowledgment policy component (not shown) to detect the existence and state of one or more packet acknowledgment attributes. In response to identifying the idle information, attribute~~

~~option handler component 564 may modify the state of the packet acknowledgment option. Thus, in response to identifying the idle information, attribute~~

~~option handler component 564 may prevent and/or alter the time an acknowledgment is sent in a packet data from second node 604 to first node 602 in the TCP connection.~~

~~[0125] As described herein an ITP header for detecting an idle time period for a TCP connection may serve a number of purposes. A first node in a TCP connection may via an ITP header inform and/or otherwise identify to a second node in the connection one or more durations for detecting an idle time period by one or both nodes. Given multiple purposes, one or more types of ITP headers may be supported and/or an ITP header may be structured to support one or more of the described services. An exchange of ITP headers may be informational and/or may be included in negotiation between two nodes included in a TCP connection. When used in a negotiation, an ITP header may be included in a negotiation protocol that has an identifiable end during a portion of the existence of a TCP connection or may be included in a negotiation that may remain ongoing throughout the existence of a TCP connection. Those skilled in the art will recognize the list of services in this paragraph is not exhaustive.~~

configured for identifying, to the program component, the selected one of the first resource and the second resource for processing. Figs. 4a-d illustrate cost operations components 408 as adaptations and/or analogs of cost operations component 308 in Fig. 3. One or more cost operations components 408 operate in an execution environment 401.

[0103] As described above, resources may be selected, based on a measure of a processing cost, in response to detecting a user input for selecting the resource. Selection information may be received by a UI element handler component 433 for a selectable representation of a resource. The UI element handler component 433 receiving the selection information may provide and/or otherwise identify the resource to cost operations component 408. Cost operations component 408 may access the resource, transform the resource into a format suitable for processing by the program component, and/or identify the resource to the program component for processing. [0104] In Fig. 4a, cost operations component 408a may identify a resource selected based on a measure of a processing cost to any one or more components in application 403a. In Fig. 4b and in Fig. 4d, cost operations component 408b and/or cost operations
44

component 408d may identify a resource selected based on a measure of a processing cost to any one or more components in browser 403b, network application agent 405b, and/or to network application 403d in Fig. 4d. Cost operations component 408b and cost operations component 408d may operate alone without the presence of the other, may operate independently while the other is operating, or may interoperate to identify the selected resource in various aspects. In Fig. 4c, cost operations component 408c

may identify a resource selected based on a measure of a processing cost to any of one or more program components including various applications 403c. Alternatively or additionally, cost operations component 408c may identify a resource selected based on a measure of a processing cost to a program

component operating in another execution environment including and/or otherwise provided by another node.

[0105] Also as described above, a resource may be identified automatically, in

response to being selected. A cost operations component 408 may access the resource, transform the resource into a format suitable for processing by the program component, and/or otherwise identify the resource to the program component for processing.

[0106] A program component may be performing an operation that includes processing a first resource. A second resource selected based on a measure of a processing cost may be identified to the program component to perform the operation instead of the first resource. For example, application 403a in Fig. 4a operating in a user device, such as user node 502 in Fig. 5, may present a resource as desktop background 604 in Fig. 6. When user node 502 is operating on a battery with an estimated energy available

45

exceeding a specified time threshold, such as one hour, cost operations component 408a may select a video resource to present in display presentation space 602 as desktop background 604 based on a measure of a processing cost determined for the video resource. When user node 502 is operating on a battery with an estimated energy available less than a specified time threshold, such as one hour, cost operations component 408a may select a still image resource to present in display presentation space 602 as desktop background 604 where a measure of a processing cost for the still image meets an energy condition based on the energy available in the battery and the video does not meet the condition.

[0107] In another aspect, a program component may be performing an operation that includes processing a first resource. A second resource selected based on a measure of a processing cost may be identified to the program component to perform the operation in addition to performing the operation including processing the first resource. For example, network application agent 405b in Fig. 4b operating in user node 502 may upload files to network application 403d operating in application provider node 506 via network 504. When network 504 is an intranet in a home or business with no monetary charge based on bandwidth utilization, a measure of processing cost based on a metric for measuring the monetary cost of bandwidth may be zero or near zero for resources to transfer from user node 502 to application provider node 506. A user may select a first resource for upload. Based on the bandwidth cost, cost operations component 408b may identify an additional resource to transmit, in parallel with the first resource, from user node 502 to application provider node 506 to transfer. Alternatively or additionally,

46

cost operations component 408d may send a matching criterion to identify one or more additional resources to receive in parallel with the first resource from network application agent 405b operating in browser 403b.

[0108] As described herein, a resource may be selected based on a measure of a processing cost. At some time after the selection, the resource may be identified to a program component in response to detecting a specified event. A number of resources may be selected based on respective measures of a

processing cost. The resources may be associated with a number of respective events. In response to a first event in the number of events, a first resource associated with the first event may be identified to a program component. In Fig. 4c, cost management subsystem 407c may change permissions, roles, etc. for selected resources based on measures of a processing cost allowing access to and/or otherwise identifying resources that match a current cost condition.

[0109] The method illustrated in Fig. 2 may include additional aspects supported by various adaptations and/or analogs of the arrangement of components in Fig. 3. Receiving resource information identifying a resource may include intercepting a communication for performing an operation, detecting an access for retrieving the resource, reading a message recording at least one of an access to the resource and a request for performing an operation, identifying a mapping identifying the type of the resource and an operation, and/or detecting a change in a program component for performing an operation, a change in the resource, and a change in a hardware component included in processing the resource. A cost advisor component 402 in Figs.

47

4a-d may be configured to interoperate with various components, to receive resource information, including a file system, a data store, a data storage device, a GUI subsystem 437, an input driver 441, a network stack 411, and an application protocol component 413, to name a few examples.

[0110] Receiving resource information may include receiving a communication, intercepting a communication, and/or initiating a communication. Receiving the resource information may include detecting an access to at least one of the first resource, the second resource, and the program component. Detecting an access may include detecting an access to at least one of a semaphore, a lock, a data storage location, a component of a input subsystem, a component of a presentation subsystem, a storage subsystem, a component of a networking subsystem, a component of a graphics subsystem, a component of an audio subsystem, a display adapter, a display device, an audio adapter, an audio output device, a tactile presentation subsystem, a tactile output device, an access control component, a serialization component, a synchronization component, a thread, an input device driver, an input device, another application, a code library, a database, a service operating in remote node via a network, text data, image data, audio data, tactile data, a message formatted according to a communication protocol, a service, a presence entity, a subscription, a software component, a hardware component, a transaction, a media stream, a location, a measuring device, data, an instruction, a persistently stored resource, a resource stored in volatile storage, a network resource, a preexisting resource, a dynamically generated

48

resource which may already exist, a service for generating the resource, a font, an encoding, a format, a mechanical resource, and an optical resource.

[0111] Image data may include a still image, a video, a background image, and/or an image for representing another resource. Audio data may include a song, a voice message, and/or a sound for indicating an event.

[0112] Exemplary hardware components that may be included in processing a resource include an IPU, an output device, a storage device, an input device, a networking component, a bus, a physical processor memory, and/or a switching fabric. [0113] A metric may be selected based on a resource, an operation, a hardware component, the program component, a user, a group, a role, a task, a time, a location, and/or a device for performing the operation and/or for providing the resource. A metric for measuring a processing cost for a resource may be based on at least one of the resource, an operation included in processing the resource, a hardware component included in performing an operation that includes processing the resource, a user, a group, a role, a task, a time, a location, and hardware for providing and/or otherwise maintaining the resource. For example, a cost monitor component 404 may monitor a

rate of energy received by a display device for an image resource presented and/or to be presented by the display device. A cost monitor component 404 may determine a measure of a processing cost according to a metric based on a count of bytes in a file resource for transmitting a resource via a network. [0114] Various aspects and adaptations of cost monitor component 304 in Fig. 3 may determine a measure of a processing cost for a metric based on a flow of electricity,

49

stored energy, mechanical resistance, electrical resistance, time, a count of a particular energy related event, money, an environmental impact, a health impact, a change in size, a change in mass and/or weight, a safety impact, heat, light, and/or movement. Correspondingly, various aspects and adaptations of cost monitor component 304 may determine a measure of electrical energy, a measure of stored energy, a measure of mechanical resistance, a measure of electrical resistance, a measure of time, a count of of a particular event, a measure of monetary cost, a measure of heat, a measure of light, a measure of distance, a measure of mass, a measure of size, and/or a measure of weight.

[0115] A processing cost may be based on a flow of electricity, stored energy, mechanical resistance, electrical resistance, time, a count of a particular event, money, a size, mass, weight, heat, light, and/or movement. A first measure and a second measure may include a measure of electrical energy, a measure of stored energy, a measure of mechanical resistance, a measure of electrical resistance, a measure of time, a count of a particular event, a measure of monetary cost, a measure of heat, a measure of light, a measure of distance, a measure of mass, a measure of size, and/or a measure of weight. A count may be based on IPU cycles, disk spins, data read operations, data write operations, refreshes of at least a portion of a presentation space, display refreshes, data transmitted via a network, data received via a network, and/or human movement. A measure of human movement may be based on a measure of dispersion of key presses; a pattern and frequency of movement of a tracking device;

and a count of at least one of key presses, squeezes, pushes, pulls, changes between lower case and upper case, and/or a count of numerical digits.

[0116] A measure of a processing cost may be determined based on a previously determined measure of a processing cost. Determining a measure may be based on locating a predefined measure based on at least one of the resource and the program component.

[0117] Determining a measure may include sending a message via a network to a node for determining the measure. A response may be received via the network including and/or otherwise identifying the measure.

[0118] In an aspect, an energy condition may be specified. A measure of a processing

cost determined for a resource may be determined for evaluating an energy condition to determine whether the energy condition is met. An energy condition may be identified for evaluating and/or may be evaluated based on an energy source, an amount of energy available, an amount of energy available in a battery and/or other energy store, an amount and/or rate of energy used and/or currently being used for processing another resource, a location of the device, and a time required for restoring an energy store to a specified state, to name a few examples.

[0119] In response to determining whether an energy condition is met for a resource, a representation of the resource presented by an output device may be presented as user selectable or not user selectable. For example, in Fig. 4c when an amount of energy available from a battery in a mobile device falls below a specified threshold, files over a specified size may not be presented or may be presented but not selectable for

51

attaching to an email by an email application represented by second application 403c2. In Fig. 4b and in Fig. 4d, one or both of cost monitor component 404b and cost monitor component 404d may receive information identifying a monetary cost of transmitting data via a current network access provider. Based on the cost, certain resources may be presented as selectable for certain operations including transmitting data in the resources and may be presented as not selectable for other network operations based on one or more cost conditions associated with the operations, the program components, and/or the resources. A first resource may be selected for presenting as user selectable, and a second resource may be not be presented or may be presented as un-selectable.

[0120] A resource may be selected from a plurality of resources in response to user selection information. For example, a first selectable representation of a first resource and a second selectable representation of a second resource may be presented to a user based on respective measures of a processing cost. Selection information may be received in response to a detected user input identifying a resource to be selected from the plurality.

[0121] A selected resource may be identified to a program component for processing instead of a resource currently being processed by the program component. A selected resource may be identified to a program component for

processing in addition to a resource currently being processed by the program component.

[0122] A resource may be identified to a program component by disabling access to other resources and allowing access to the resource identifying the resource to the

52

program component. A cost operations component 408 in Figs. 4a-d may be configured to enable and/or disable access to resources selected by a corresponding cost director component 406.

[0123] Identifying a selected resource to a program component may include providing for terminating processing of a resource currently being processed by the program component. Terminating processing of the resource may include terminating processing of the program component. Subsequently, processing of the selected resource by the program component may be initiated. Initiating processing of the selected program component may include initiating and/or restarting operation of the program component.

For example, cost operations component 408c in Fig. 4c may restart application 403c1 identifying a selected resource for processing by restarted application 403c1.

[0124] In an example, a first resource may include first image data, the second resource may include second image data, and the program component may be configured to present image data in a presentation space of a display allocated for presenting a desktop image. An energy condition may be detected during presentation of the first image data in the presentation space by the program component. The second resource may be selected based on a first measure of a processing cost for the first resource and a second measure of the processing cost for the second resource.

The selection may be in response to detecting the condition. In response to the

selection, the second resource may be identified to the program component to present the second image data in the presentation space.

53

[0125] In various aspects of the method illustrated in Fig. 2, at least one of receiving the resource information, determining at least one of the first measure and the second measure, selecting one of the first resource and the second resource, and identifying the selected one of the first resource and the second resource may be performed in response to at least one of detecting a change in a first energy source, receiving energy from a first energy source then receiving energy from a second energy source,

detecting that energy is flowing to a first source increasing the amount of energy stored in the first source, and detecting a change in an amount of energy available from a first energy source. For example, cost management subsystem 407c in Fig. 4c may detect a change in receiving energy from an electrical

outlet to receiving energy from a battery in execution environment 401c. In response to detecting the change, cost director component 406c may select a resource based on a measure of a processing cost. In response to detecting the change, cost operations component 408c may identify a selected resource to a program component for processing.

[0126] In various aspects of the method illustrated in Fig. 2, at least one of receiving the resource information, determining at least one of the first measure and the second measure, selecting one of the first resource and the second resource, and identifying

the selected one of the first resource and the second resource may be performed in

response to a change in at least one of a monetary cost of energy, an organization providing energy, a rate of energy utilization, a utilization time of a first energy source, a user, a geospatial location, heat, light, and a component for at least one of storing, transmitting, and receiving energy. For example, in Fig. 4c, a cost management

54

subsystem 407c may receive event information identifying one or more of listed events and/or conditions in directing performance of the method illustrated in Fig. 2 by the arrangement of components illustrated in Fig. 4c.

[0127] In another example, cost director component 406a in Fig. 4a may detect a change in an amount of energy. In an aspect, energy may be flowing from a first energy source to one or more hardware components. Subsequently, energy may be detected flowing to the first energy source. Cost director component 406a may select a resource from a plurality of resources based on a measure of a processing cost for one or more of the respective resources in the plurality. Alternatively or additionally, cost operations director component 408a may identify a selected resource to a program component for processing in response to detecting a change in energy received from an energy source.

[0128] This application is related to the following which are each incorporated herein by reference in their entirety for all purposes: U.S. Patent Application No. 14/294,059, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SELECTING A RESOURCE BASED ON A MEASURE OF A PROCESSING COST,"

filed 06-02-2014 which in turn claims priority to U.S. Patent Application No. 12/857,851,

entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SELECTING A RESOURCE BASED ON A MEASURE OF A PROCESSING COST,"

filed 08-17-2010; U.S. Patent Application No. 13/941,502, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A NON-RETURNABLE COMMAND RESPONSE BASED ON A MARKUP ELEMENT," filed 07-

55

14-2013 which in turn claims priority to U.S. Patent Application No. 12/789,550, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PROCESSING A NON-RETURNABLE COMMAND RESPONSE BASED ON A MARKUP ELEMENT," filed 05-28-2010; U.S. Patent Application No. 13/477,402, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION," filed 05-

22-2012 which in turn claims priority to U.S. Patent Application No. 12/714,454, entitled "METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION, filed 02-27-2010; U.S. Patent Application No. 12/857,847, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PRESENTING AN INDICATION OF A COST OF PROCESSING A RESOURCE," filed 08-17-2010; U.S. Patent Application No. 12/857,857, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR SELECTING A RESOURCE IN RESPONSE TO A CHANGE IN AVAILABLE ENERGY," filed on 08-17-2010; U.S. Patent Application No. 12/857,836, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR MAINTAINING A RESOURCE BASED ON A COST OF ENERGY," filed 08-17-2010; U.S. Patent No. 8,346,853, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUGTS FOR PROGESSIONG AN ATTACHED COMMAND RESPONSE," filed 05-27-2010; U.S. Patent Application No. 12/788,373, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PREVENTING PROCESSING OF AN HTTP RESPONSE," filed 05-27-2010; U.S. Patent Application No. 12/789,538, entitled "METHODS, SYSTEMS, AND PROGRAM

56

PRODUCTS FOR PROCESSING AN ATTACHED COMMAND RESPONSE BASED

ON A MARKUP ELEMENT," filed 05-28-2010; U.S. Patent Application No. 12/788,381,

entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A

COMBINED COMMAND RESPONSE," filed 05-27-2010; U.S. Patent Application No. 12/789,568, entitled "METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR PROCESSING A COMBINED COMMAND RESPONSE BASED ON A MARKUP ELEMENT," filed 05-28-2010; and U.S. Patent Application No. 12/714,063, entitled

"METHODS, SYSTEMS, AND PROGRAM PRODUCTS FOR DETECTING AN IDLE

TCP CONNECTION," filed on 02-26-2010. It should be noted that any feature disclosed in the context of any embodiment of this and/or any of the foregoing applications may be incorporated in the context of any embodiment of this and/or any of the foregoing applications.

[0129] To the accomplishment of the foregoing and related ends, the descriptions and annexed drawings set forth certain illustrative aspects and implementations of the disclosure. These are indicative of but a few of the various ways in which one or more aspects of the disclosure may be employed. The other aspects, advantages, and novel features of the disclosure will become apparent from the detailed description included herein when considered in conjunction with the annexed drawings.

[~~0126~~0130] It should be understood that the various components illustrated in the various block diagrams represent logical components that are configured to perform the functionality described herein and may be implemented in software, hardware, or a combination of the two. Moreover, some or all of these logical components may be

57

combined, some may be omitted altogether, and additional components may be added while still achieving the functionality described herein. Thus, the subject matter described herein may be embodied in many different variations, and all such variations are contemplated to be within the scope of what is claimed.

[~~0127~~0131] To facilitate an understanding of the subject matter described above, many aspects are described in terms of sequences of actions that may be performed by elements of a computer system. For example, it will be recognized that the various actions may be performed by specialized circuits or circuitry (e.g., discrete logic gates interconnected to perform a specialized function), by program instructions being executed by one or more instruction processing units, or by a combination of both. The description herein of any sequence of actions is not intended to imply that the specific order described for performing that sequence must be followed.

[~~0128~~0132] Moreover, the methods described herein may be embodied in executable instructions stored in a computer readable ~~storage~~ medium for use by or in connection with an instruction execution machine, system, apparatus, or device, such as a computer-based or processor-containing machine, system, apparatus, or device. As used ~~herein~~here, a "computer readable ~~storage~~ medium" may include one or more of any suitable media for storing the executable instructions of a computer program in one or more of an electronic, magnetic, optical, ~~and~~ electromagnetic, and infrared form, such that the instruction execution machine, system, apparatus, or device may read (or fetch) the instructions from the computer readable ~~storage~~ medium and execute the instructions for carrying out the described methods. A non-exhaustive list of conventional exemplary computer readable ~~storage~~ media 58

includes a portable computer diskette; a random access memory (RAM); a read only memory (ROM); an erasable programmable read only memory (EPROM or Flash memory); optical storage devices, including a portable compact disc (CD), a portable digital video disc (DVD), a high definition DVD (HD- DVD.TM.), and a Blu-ray.TM. disc; and the like.

[~~0129~~0133] Thus, the subject matter described herein may be embodied in many different forms, and all such forms are contemplated to be within the scope of what is claimed. It will be understood that various details may be changed without departing from the scope of the claimed subject matter. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation, as the scope of protection sought is defined by the claims as set forth hereinafter together with any equivalents ~~thereof~~ entitled to.

[~~0130~~0134] All methods described herein may be performed in any order unless otherwise indicated herein explicitly or by context. The use of the terms "a" and "an" and "the" and similar referents in the context of the foregoing description and in the context of the following claims are to be construed to include the singular and the plural, unless otherwise indicated herein explicitly or clearly contradicted by context. The foregoing description is not to be interpreted as indicating that any non-claimed element is essential to the practice of the subject matter as claimed.

