

Independent Claims 1 and 7	
[1.0] A system for providing data to a plurality of compute nodes, the system comprising:	[7.0] A method for providing data to a plurality of compute nodes, comprising:
[1.1] a first storage unit configured to store blocks of a root image of said compute nodes;	[7.1] storing blocks of a root image of said compute nodes on a first storage unit;
[1.2] a plurality of second storage units configured to store leaf images of respective compute nodes, said leaf images including only additional data blocks not previously contained in said root image and changes made by respective compute nodes to the blocks of said root image, wherein said leaf images of respective compute nodes do not include blocks of said root image that are unchanged by respective compute nodes; and	[7.2] storing leaf images for respective compute nodes on respective second storage units, said leaf images including only additional data blocks not previously contained in said root image and changes made by respective compute nodes to the blocks of the root image, wherein said leaf images of respective compute nodes do not include blocks of said root image that are unchanged by respective compute nodes; and
[1.3] a cache configured to cache blocks of said root image previously accessed by at least one of said compute nodes.	[7.3] caching blocks of said root image that have been accessed by at least one of said compute nodes in a cache memory.

Dependent Claims 2-6 and 8-13	
[2] The system as recited in claim 1 wherein said cache is configured to store X most recently accessed blocks of said root image, and wherein X represents a cache threshold value.	
[3] The system as recited in claim 1 wherein said first storage unit, said second storage units, and said cache are contained within a single storage appliance.	
	[8.1] The method as recited in claim 7 further comprising: receiving a read request from at least one of said compute nodes, wherein a first portion of the data requested is currently stored in said cache memory; and providing said first portion of said data to said at least one of said compute nodes from said cache memory.
	[8.2] providing said first portion of said data to said at least one of said compute nodes from said cache memory.
	[9] The method as recited in claim 8 further comprising:

	updating said cache memory based on said read request.
	[10.0] The method as recited in claim 9 wherein a second portion of the data requested is not currently stored in said cache memory and said updating comprises:
	[10.1] caching said second portion in said cache memory; and
	[10.2] removing the least recently accessed data from said cache memory if the amount of data in said cache memory is above a threshold value.
[4] The system as recited in claim 1 further comprising: a plurality of union block devices configured to interface between respective compute nodes and said first storage unit, respective second storage units, and said cache to distribute application environments to the compute nodes, wherein said union block devices are configured to create said application environments by merging the blocks of said root image	[11] The method as recited in claim 7 further comprising: merging the blocks of said root image with the blocks of respective leaf images to create cohesive respective application environments.

with the blocks of respective leaf images.	
[5] The system as recited in claim 4 wherein said union block devices comprise low-level drivers for interfacing between the file systems of respective compute nodes and said first storage unit, respective second storage units, and said cache.	[12] The method as recited in claim 11 wherein said merging occurs at an operational level between file systems of the respective compute nodes and said first storage unit, respective second storage units, and said cache memory.
[6] The system as recited in claim 1 wherein said first storage unit is read-only.	[13] The method as recited in claim 7 wherein said first storage unit is read-only.

Independent Claims 14, 19, and 23		
[14.0] A system for indexing file systems for a plurality of compute nodes, the system comprising:	[19.0] A method for indexing file systems for a plurality of compute nodes, comprising:	[23.0] A computer-readable storage medium having instructions stored thereon that, in response to execution by at least one computing device, cause the at least one computing device to:
[14.1] a first storage unit configured to store blocks of a root image of said compute nodes;	[19.1] storing blocks of a root image of said compute nodes on a first storage unit;	[23.1] receive data blocks of a file system, said data blocks comprising a root image portion
[14.2] a plurality of second storage units configured to store leaf images of respective compute nodes, said leaf images comprising only additional data blocks not previously contained in said root image and changes made by respective compute nodes to the blocks of said root image, wherein said leaf images of respective	[19.2] storing leaf images for respective compute nodes on respective second storage units, said leaf images comprising only additional data blocks not previously contained in said root image and changes made by respective compute nodes to the blocks of the root image, wherein said leaf images for respective compute nodes do not	[23.2] and leaf image portion, said leaf image portion comprising only additional data blocks not previously contained in said root image portion and changes made by said first compute node to the blocks of said root image, wherein said leaf image portion does not include blocks of said root image that are

compute nodes do not include blocks of said root image that are unchanged by respective compute nodes; and	include blocks of said root image that are unchanged by respective compute nodes;	unchanged by said first compute node,
[14.3] a plurality of union block devices corresponding to said compute nodes, said union block devices configured to interface between said compute nodes and said first and second storage units to distribute said file systems to said compute nodes, wherein said union block devices are configured to create said file systems by merging the blocks of said root image stored on the first storage unit with the blocks of respective leaf images stored on	[19.3] merging the blocks of said root image with the blocks of respective leaf images stored on respective second storage units to create respective file systems for respective compute nodes;	[23.3] wherein said file system is the result of merging said root image portion and said leaf image portion together at the block-level;

respective second storage units, and		
[14.4] wherein further at least one of said compute nodes is configured to index said root image and provide the indexing results to another of said compute nodes.	[19.4] receiving indexing results pertaining to said root image from one of said compute nodes; and providing said indexing results to the others of said compute nodes.	[23.4] index said root image portion; and provide the results of said indexing to a second compute node, wherein said logic encoded in the one or more tangible media comprise computer executable instructions executed by the first compute node.

Dependent Claims 15-18, 20-22, and 24-27		
[15] The system as recited in claim 14 wherein said first storage unit and said second storage units are contained within a single storage appliance.		
	[20] The method as recited in claim 19 further comprising: storing said indexing results on a shared storage unit.	[24] The computer-readable storage medium of claim 23, wherein the instructions further cause the at least one computing device to store said results of said indexing on a shared storage unit accessible by said second compute node.
[16] The system as recited in claim 14 further comprising: a plurality of union block devices configured to interface between respective compute nodes and said first storage unit		

and respective second storage units, said union block devices configured to distribute application environments to the compute nodes, wherein said union block devices are configured to create said application environments by merging the blocks of said root image with the blocks of respective leaf images.		
		[25] The computer-readable storage medium of claim 23, wherein the instructions further cause the at least one computing device to index said leaf image portion.
		[26] The computer-readable storage medium of claim 25, wherein the instructions further cause the at least one

		computing device to re-index said file system by re-indexing said leaf image portion and merging the results of said re-indexing of said leaf image portion with said results of said indexing of said root image portion.
[17] The system as recited in claim 16 wherein said union block devices comprise low-level drivers for interfacing between the file systems of respective compute nodes and said first storage unit, respective second storage units, and said cache.	[21] The method as recited in claim 19 wherein said merging occurs at an operational level between respective file systems of the compute nodes and said first storage unit and respective second storage units.	
[18] The system as recited in claim 14 wherein said first storage unit is read-only.	[22] The method as recited in claim 19 wherein said first storage unit is read-only.	[27] The computer-readable storage medium of claim 23 wherein said root image portion is read-only.