
From: "Srinivas Varadarajan" <srini@vigyanlabs.com>
Sent: Sun 9/7/2008 11:44:16 AM (UTC-07:00)
To: "Badri S" <S.Badrinarayanan@compassites.net>, "Seymour Duncker" <seymour.duncker@icharts.net>, "Girish Karachiwala @ Compassites" <girish.karachiwala@compassites.net>
Cc: mukund.krishnamurthy@compassites.net
Subject: System Sizing and Initial Capacity Plan document
Attachment: iChartsSystemSizingv1.0.pdf

Hi All,

Attached here is the first version of the System Sizing and the Initial Capacity Plan document.
Thank you for all the support.

With Regards,
Srinivas



Research + Innovation + IT + Consulting

iCharts System Sizing Report

Version 1.0, 7th September 2008

Compassites and VIGYANLABS Confidential

Table of Contents

1 Background!	1
Introduction!	1
Scope!	1
Intended Audience!	1
References!	1
2 Workload Model!	2
Overview!	2
Methodology!	2
Workload Pattern!	2
Performance Testing!	4
Analysis of Performance Test Results!	5
<i>Bandwidth Usage!</i>	5
<i>CPU Usage!</i>	6
3 Deployment Architecture!	7
Deployment Model!	7
Performance and Scalability!	7
Scaling Web Tier!	8
Scaling Application Tier!	8
Scaling Database Tier!	8
<i>Master and Backup Slave!</i>	9

Cache Servers!	10
Availability and Fail Over or Redundancy!	10
Web Server Failure!	10
Application Server Failure!	10
Database Failure!	10
System Security!	10
System Monitoring and Capacity Planning!	11
Initial Deployment Hardware!	11

1 Background

Introduction

This document describes the deployment architecture for the icharts application. The deployment architecture is based on the anticipated workload pattern, availability requirements, infrastructure facilities available from the service provider and finally the resource usage of the application based on the initial performance test results.

Scope

The document covers the deployment architecture, system monitoring and capacity planning for the iCharts application.

Intended Audience

This document is intended for iCharts team, managers and technical experts.

References

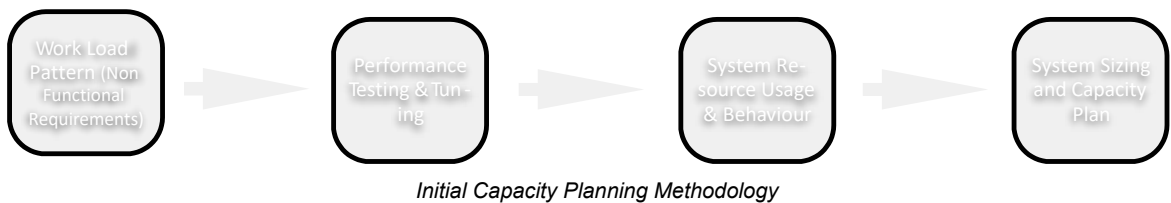
1. iCharts Architecture Document
2. iCharts Non-Functional Requirements (spreadsheet)
3. iCharts Application Performance Test Results

2 Workload Model

Overview

The workload pattern is based on the anticipated usage of the system by the users. The real life workload pattern is likely to change over time and should be continuously monitored to fine tune the system. Based on the anticipated workload pattern a few important scenarios were chosen for the performance tests and the system resource usage was analyzed, the system resource usage for these scenarios forms the basis for the model.

Methodology



Workload Pattern

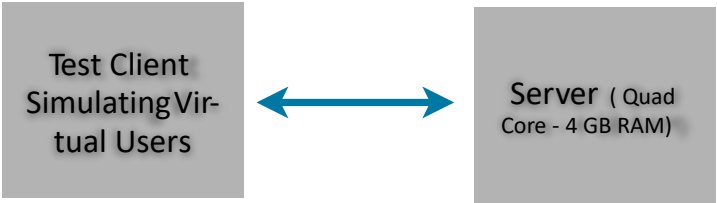
The table below shows the anticipated usage pattern. Max concurrent user have been derived by assuming a 10 hour spread for the load and 10% of this load as the concurrent users (For example for 5000/day - 500 per hour, - 10% of 500 - 50 Concurrent)

Use Case name/ref	Brief description of the use case (what the end user does)	Frequency of usage of this use case	Max Concurrent users
Charts Tab Listing	User visits the icharts.net portal and the first tab that he/ she sees is the chart listing page.	5000/ day	50
Charts Tab Filter	User can filter the charts seen in listing page to view most recent, recently modified, todays modified, this weeks modified charts	5000/ day	50
Charts page navigation through pagination	User can navigate to next or previous set of charts using navigation links	5000/ day	50
Chart detail page visit	User can click on thumbnail to see the detail of the chart	5000/ day	50
Add to favourite	Logged in user can add a chart in his favourites list	1000/ day	10
Chart Search	User can search the chart using text search which searches the chart based on 10 different criteria.	500/ day	5
Community Page listing	User can navigate to Community listing page which shows the listing of users who have opted to make their profiles public	500/ day	5
Charts page navigation through pagination	User can navigate to next or previous set of u users using navigation links	1000/ day	10

User profile page visit	User can see complete public profile of the user	100/ day	1
User Registration	User registers with the application. An email is sent to user after registration. Upon confirmation by clicking link, user is registered.	1000/ month	
User Login	User can login to the system using his/. Her credentials	5000/ day	50
My Charts page listing	Once user logs in, he/ she is taken to the My Charts tab which is the gallery of charts and data sheets created by user	5000/ day	50
Create New Chart	User can click "Create New Chart" button to create a new chart. This would take the user to flex based studio where user can create the chart.	500/ day	5
Edit Chart	User can click on edit icon below the chart thumbnail in My Charts listing to edit the chart. This takes user to ichart studio where he/ she can edit the chart.	500 / day	5
Delete Chart	User can click on delete icon below thumbnail to delete the his/ her chart.	20 / week	
Share Chart	User can click share ichart icon below the thumbnail to share the ichart with other people using email ids.	500 / day	5
Duplicate Chart	User can click duplicate ichart icon below the thumbnail to create the duplicate of the chart	100 / day	
Data sheet listing page	User can view the listing of data sheets he/ she has created	1000/ day	10
Chart listing page filter	User can filter the listing of the charts using likes like 'Published charts', unpublished charts etc.	1000/day	10
Data Sheet Detail Page	User can go to detail page of data sheet where listing of related charts etc are shown.	1000/ day	10
Edit/ Add Data Sheet	User can click edit data sheet icon below the thumbnail to edit the data sheet in flex studio	250 / day	2
Create folder	User can create folder to arrange the charts	200/ week	
Add chart to folder	User can add a chart to folder using drag/drop function,	10/ week	
Delete folder	User can delete the folder he/ she created.	5/ week	
Spotlight chart	User can select a chart to be spotlighted	1/week	
spotlight user	User with privilege can select a user to be spotlighted	1/week	
Publish chart/ republish chart	User can publish a chart to be shown in ichart portal or to be embedded in htmls	100/day	1
Unpublish chart	User can decide to unpublish the chart	40/ week	
Download chart	User can download a chart in gif, tiff, jpeg, swf format.	500/ day	5
Add comment	User can add a comment on chart	100/ week	
feature chart	User with privilege can set a chart as featured	once a week	
	Max Simultaneous Users		384

Performance Testing

The performance test environment consists of a desktop machine simulating virtual users as per the workload pattern. The Web, Image Server, Application Server and MySQL Server are all hosted on a Virtualized Quad Core (Intel) Machine with 4 GB Ram. The Test Client and the Server are interconnected by a 100Mbps Switched Network.



Work Load Description	Action	System Behaviour
Anonymous Users view the home page (chart tab listing), view few charts and exit	25 Simultaneous Virtual users with no delay perform the same action repetitively	Highly network Intensive, Moderate CPU, Low I/O
User Logs in, accesses the home page, clicks on myCharts page listing	25 Simultaneous Virtual users with no delay perform the same action repetitively	Highly network Intensive, Moderately CPU Intensive (database)
User logs in, create a new chart, adds title, saves the chart, accesses folders, previews the created chart and logs out	25 Simultaneous Virtual users with no delay perform the same action repetitively	Moderate CPU Intensive (java), Moderate Network Intensive
New User Registers on the Portal	25 Simultaneous Virtual users with no delay perform the same action repetitively	Highly CPU Intensive (java), Moderately Network Intensive, Low I/O

Analysis of Performance Test Results

In this section we look at a set of samples from the performance test results which form the basis for the sizing.

Bandwidth Usage

As shown in the figure below, there is a significant usage of bandwidth when the users access the portal the first time as their browsers

data, the pie chart

the ideal situation

cached. The primed

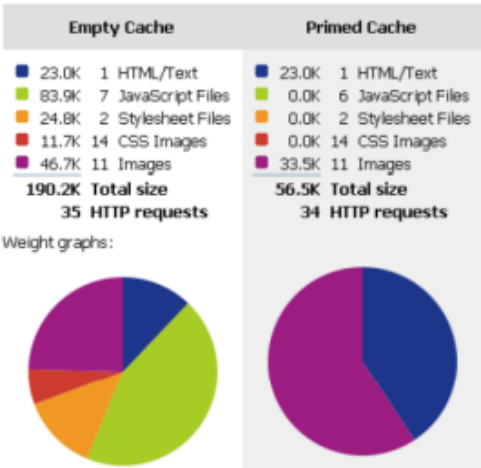
when most of the

amount of content

frequently the

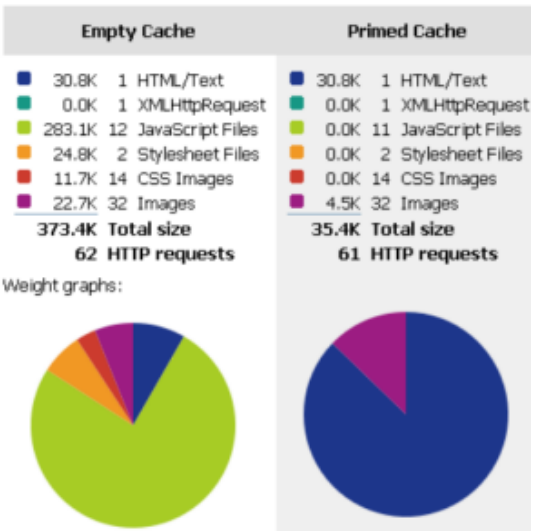
and any other

the site.



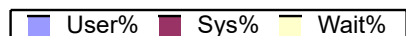
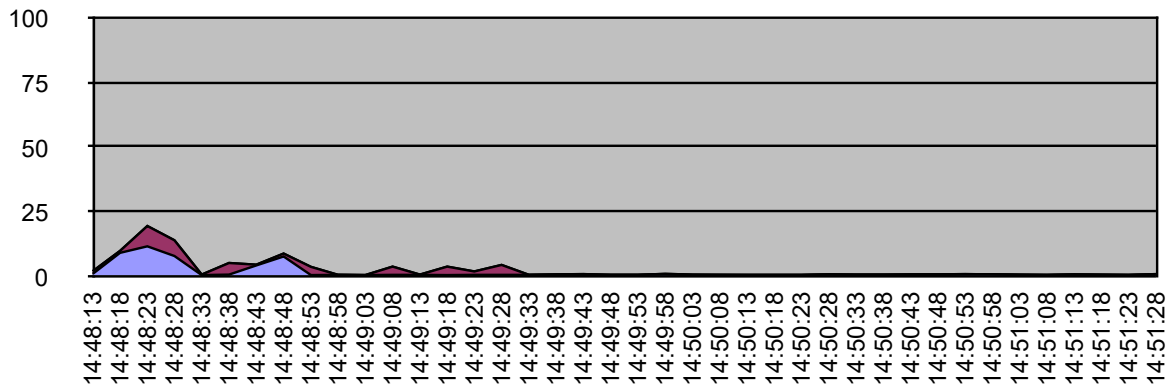
Bandwidth Used while accessing the home page (no login, with compression enabled)

Note that there is a significant amount of bandwidth used when the users logs in to the access the my gallery, images, style sheets get downloaded is significantly

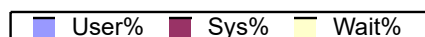
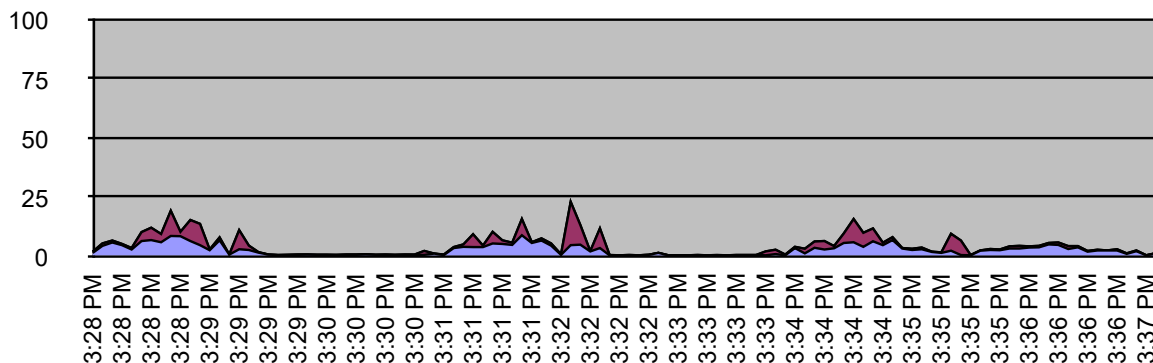


amount of bandwidth used portal for the first time and subsequently the java scripts, cached and the content reduced.

Bandwidth used during login and access to mygallery (with compression enabled)

CPU Usage**CPU Total icharts 9/6/2008***CPU Usage for 25 Simultaneous Users during Registration Process*

Note that in the chart above, nearly 25% of the CPU is consumed (in a 4 CPU system) when there are 25 simultaneous requests, a maximum of 75 simultaneous registrations are possible on a 4 CPU box.

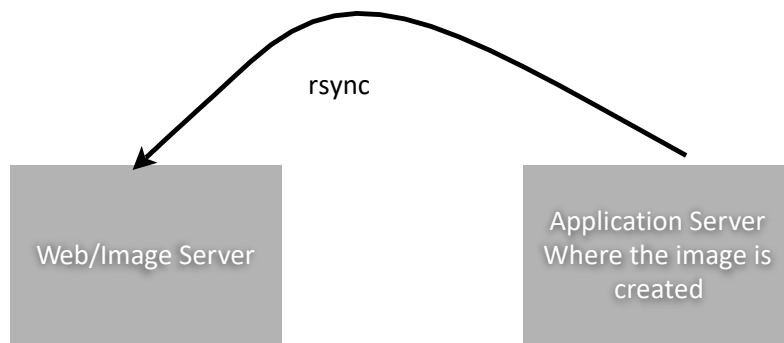
**CPU Total icharts 9/6/2008***CPU Usage for 10 Simultaneous Users during login and chart creation*

The chart above shows the CPU usage when 10 Simultaneous users login, create a new chart, save and exit. Note that there are CPU spike during these activities. a 4 CPU box is likely to support between 35-50 simultaneous such operations.

3 Deployment Architecture

Deployment Model

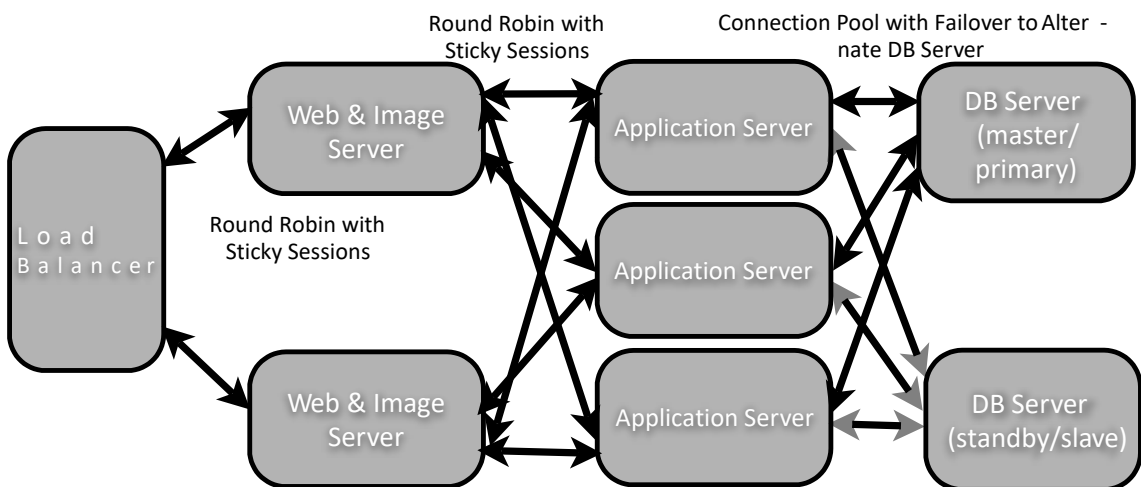
The application uses a typical MVC design pattern and is suited for a 3 tier architecture. The other aspects to be considered during deployment is the storage of images on the image server. The images are created by the application and the image files have to be accessible to the image server. To cater to a scalable model when there are multiple application servers and images servers hosted, it is necessary to move the images to the image server from the application server where it is created. Rsync is one of the widely used tools for synchronizing content/files across multiple systems.



Synchronization of Image files from application server(s) to image servers.

Performance and Scalability

As the Web application uses the standard MVC architecture, the web and the application servers can be scaled horizontally with the use of a load balancer (round robin with sticky sessions).



Logical Deployment View

To a large extent the size and number of servers are determined by the number of simultaneous requests, the type of requests, the database size (number of images or charts) and the number of registered users.

Scaling Web Tier

The Web tier handles the http connections, the image download and also serves the static files. The number of servers in the web tier depends on the following factors:

- Peak Page Visits/day - Affects the CPU and bandwidth
- Mycharts Page listing, Chart Views and Downloads/Day - Affects the CPU usage and Network Bandwidth
- Number of Image Files - Affects Cache and memory usage
- Extent of compression - Affects the CPU usage, Memory and Network Bandwidth
- First Time Visitors - Affects the CPU Usage, Memory and Network Bandwidth

Currently as per the tests with 25 simultaneous users using Chart Views the peak network bandwidth used is 30 Mbits/Sec. Similarly with 25 simultaneous users visiting the home page the peak network bandwidth used is 12 Mbits/Sec. To reduce the network bandwidth usage it is recommended that compression (mod deflate) is implemented in the Web/Image Servers, this is likely to increase the CPU usage. Usage of compression reduced the bandwidth usage by 30% and at the same time increased the CPU usage by 30% (with 25 simultaneous users). During the initial launch of the portal most of the users will be first time users and hence their browser caches are likely to be empty which means all the images files, html pages, css and other files will be downloaded. Subsequent access to the portal from the same users will result in lesser load on the server (assuming the content has not changed significantly) and also lower bandwidth usage due to the browser cache. In the long run depending on the usage of the portal and the content uploaded into it, it may be prudent to work with the network service providers who can cache the content. Over time as the number of images/charts increase, the I/O on the image servers is also likely to increase and may warrant additional memory and disks.

Scaling Application Tier

The Application tier manages the dynamic content, creation of charts and all other application logic. The number of servers in the application tier depends on the following:

- Active Sessions - How many users are currently logged in (they consume session memory)
- Mycharts Page listing, Chart Views and Downloads/Day - Affects the Application Server CPU usage (JVM)
- Studio Operations (create charts, edit, modify, delete, export etc.)/Day - Affects the Application Server CPU usage, memory usage and I/O

Currently as per the tests with 25 simultaneous users using Chart Views the peak CPU used is 35%. Similarly with 25 simultaneous users visiting the home page the peak CPU used is 10%.

Scaling Database Tier

The Database tier holds all persistent data other than the image files. The scalability of the database tier depends on the following factor:

- Mycharts Page listing, Chart Views and Downloads/Day - Affects the Database Server CPU usage and I/O

VIGYANLABS

- Studio Operations (create charts, edit, modify, delete, export etc.)/Day - Affects the Database Server CPU usage, memory usage and I/O
- User Registrations - Affects the Database Server CPU Usage and I/O
- Number of Charts, Registered Users - Affects the database Storage, I/O and CPU

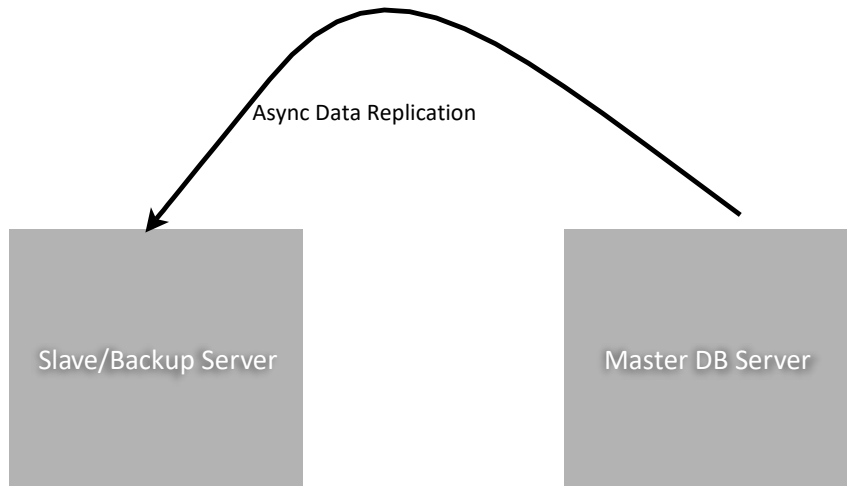
Currently as per the tests with 25 simultaneous users doing operations like ChartViews, Visiting the home page, Chart Creation the CPU usage was 10%. I/O can be scaled by using a Raid array with multiple disks.

The storage requirements at present is 100 MB per 1000 Charts. The I/O utilization is likely to increase as the number of charts in the system increases during which time only a limited number of elements will be cached.

Unlike the Web and Application Server tiers the Database tiers cannot be scaled easily. There are multiple techniques to scale the DB tier.

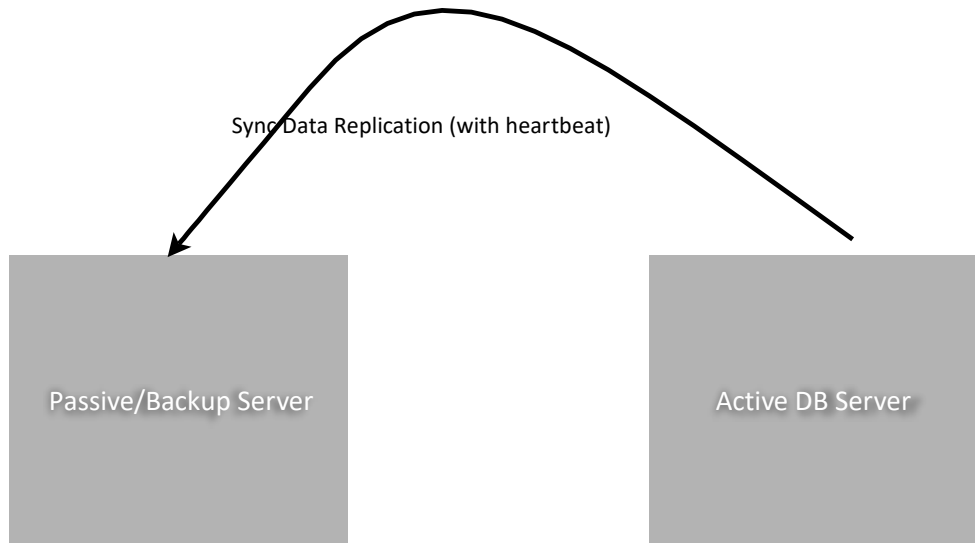
Master and Backup Slave

As shown in figure below the Master DB is primarily used for all the data management, the data is replicated to the slave server asynchronously and the database backup operations can be carried out on the slave server. The slave server can be of a lesser configuration than the master server.



Database Server Scalability with Async Replication

There is another configuration which allows for Synchronous replication with a standby node. The Active and the Passive/Backup Node are also interconnected by a heartbeat network. This mechanism provides lesser down time than the previous approach but requires similar disk configuration and a reliable network (very low latency) between the two machines. Writes in this case are much slower compared to the earlier approach.



Database Server Scalability with Sync Replication

Cache Servers

To enhance scalability further, additional cache servers can be added in future, these cache servers will sit between the Application Servers and the Database and aid in reducing the load on the database servers.

Availability and Fail Over or Redundancy

Web Server Failure

The load balancer monitors the availability of the webserver and on failure of any web server it re-directs the requests to the alternate web server.

Application Server Failure

The web servers monitor the availability of the application servers (plugins), on failure of any application server the subsequent requests are redirected to an alternate application server.

Database Failure

The database data storage is configured for high availability by means of Raid 5 (striping + mirroring), a single disk failure can be tolerated. In case of unforeseen failure of the database servers, the jdbc drivers from the application servers switch to the alternate server automatically. The alternate servers database can have a lag of few seconds to many minutes depending on the database size and the load.

System Security

The system security is ensured by shielding all the servers from the internet by means of a firewall (provided by the Service Provider). All ports other than the port 80, (and other port used for image server) should be blocked for access. The web, application and database processes should be configured to run as non-administrative with user privileges (non-root). The default login ids in all the servers (apache, jBoss and MySQL) should be disabled. In addition periodic password policy change should be implemented on all the servers. It is also recommended to implement https for the login page and the registration

pages to start with. In addition tool like Google Analytics should be used to regularly track the browser type, usage pattern of the system and identify spams if any.

System Monitoring and Capacity Planning

It is necessary to monitor the system to determine the usage of the system and to plan for additional capacity. It is recommend that the following parameters are monitored regularly to take steps for corrective action.

Parameter	Method
Unique Visitors/Page	Google Analytics
Page Hits/Sec	Google Analytics
Bandwidth Usage	Google Analytics
Page Hits/Sec for the top 10 URLs and also the Bandwidth Usage for the top 10 URLs	Google Analytics
SLte Availability Monitoring	Tools provided by the Service Provider
URL Availability Monitoring	URL Monitoring tool provided by the Service Provider
Web Server CPU, Memory, Network, I/O Usage and Disk Usage	System Tools on the server provided by the Service Provider
Application Server CPU, Memory, Network, Disk and I/O Usage	System Tools on the server provided by the Service Provider
Database Server CPU, Memory, Network, Disk and I/O Usage, Cache Hit Ratio, Database Size, Number of Rows in each of the tables (table growth rate)	System Tools on the server provided by the Service Provider

Initial Deployment Hardware

The initial deployment hardware is sized based on the anticipated load pattern and also with some head room based on the likely number of simultaneous users on the system during the launch. Further fine tuning of the application and the associated infrastructure is essential as the real usage is likely to differ from the estimates. The current estimates are based on the initial user base of about 1500 logged in with a concurrency of about 150 users (simultaneous), the actual numbers may vary based on the actual workload pattern

- Load Balancer (redundant setup provided by the service provider) - Balances the load across two web/image servers and also handles failures, uses sticky sessions to bind a logged in/active user to a particular server.
- Web Server - 2 Nos, 1-2 CPU (Dual Core) , 1 GB RAM, 250 GB Hard Disk - Each first time client is likely to have upto 15 simultaneous connections to the web server (for download of images etc.), compression of the pages is likely to consume 1% CPU (single) for each simultaneous user. In addition the periodic rsync process is likely to consume about 5-10% cpu and is likely to increase as more content is added to the system. Each Web Server is likely to have a load of 40-50% for a concurrent usage of 75-100 users, the CPU usage is also likely to decrease as the number of first time users reduce. The web servers have an approximate additional head room of 20-40%. Note that in the case of a single web server failure all

VIGYANLABS

the load on the failed web server will shift to the other active web server, hence each of the web server should have enough headroom to handle failures.

- Application Servers - 3 Nos - 1 Quad Core (or 2 dual core) CPU, 4 GB RAM, 250 GB Hard Disk. This is based on the workload assumption that (as tested in the lab) each Application Server can handle a load of upto 50 simultaneous users (50-60% accessing the home page, 20-30% who login and go to my gallery, 5-10% who create charts and 5-10% users doing other operations like registration etc.). The actual number of simultaneous users is likely to vary based on the actual workload pattern. The three servers are required to handle a combined load of 150+ simultaneous user and an additional number of logged in users (upto 2000 across the servers). In addition it is estimated that the rsync process which synchronizes the content across the servers is likely to use from 5-10% of CPU (this may vary based on the number of images created/modified/deleted).
- Database Servers - Primary - 1 No - 2 CPU (Quad Core), 8 GB RAM, 2 * 73 GB RAID 1 for OS, 4 * 73 GB RAID 5 for data - Backup - 1 CPU, 1 GB RAM, 1 * 250 GB Disk. The database load observed during the testing was fairly low, less than 10 % for 25 concurrent users, the CPU usage and I/O are likely to increase with increase in number of users, increased image/charts and the usage pattern. In case of a failure of the primary database server, the backup database server should be able to handle the current load (150 simultaneous users) with 20-30% increased response time. In case of a failure of the primary server it is necessary to restore the primary back at the earliest.