WCMS Architecture

January 6, 2010

Author: Charles McIntyre
This document is for the implementation of the Campus WCMS Service (WCMS). This narrative is addressing initial infrastructure design only, not any application configuration which will be documented separately.

Please note that the Hannon Hill vendor does not provide detailed infrastructure whitepapers nor has an infrastructure support group. This technical design represents our collective best effort in architecture for the cascade server product, which we have no experience with or training. The majority of details come from previous similar implementations and discussions with other higher education institutions that have previously implemented Cascade Server.

1. Project Proposal

1.1. Implementation of Campus WCMS

1.2. See WCMS Phase II Project Charter


2. Requirements

2.1. See Campus WCMS Requirements


2.2. Non-Functional Requirements

2.2.1. Historical Data

2.2.1.1. Sites (Quantity x Usage trends 3-5 years prior)

2.2.1.1.1. WWW = 25 sites

2.2.1.1.1.1. Trending is flat to negative, with some sites being migrated off-site or onto www2

2.2.1.1.2. WWW2 = 240 sites

2.2.1.1.2.1. Trending is slowly up, with about 3 new sites a month

2.2.1.2. Users (Role x Quantity x Usage trends 3-5 years prior)

2.2.1.2.1. Application Administrators = ????

2.2.1.2.2. Site Administrators/Content Authors = ????

2.2.1.3. Storage (Type x Quantity x Usage trends 3-5 years prior)

2.2.1.3.1. WWW = 50GB (prod & dev content)

2.2.1.3.1.1. Trending static

2.2.1.3.2. WWW2 = 65GB (prod & dev content, not including DBs)

2.2.1.3.2.1. Trending slightly up

2.2.2. Performance Characteristics

2.2.2.1. Sites (Quantity x Usage projections 3-5 years forward)

2.2.2.1.1. Initial year = 3 sites, with ???? pages and ???? objects per site

2.2.2.1.2. Yearly growth = +? sites, with ???? pages and ???? objects per site

2.2.2.2. Users (Role x Quantity x Usage projections 3-5 years forward)

2.2.2.2.1. Application Administrators

2.2.2.2.1.1. 1 primary and 2 backup app administrators for the web and app hosts

2.2.2.2.1.2. anticipating 0% change per year and 0% increase over the next 3 years

2.2.2.2.2. Site Administrators

2.2.2.2.2.1. TBD

2.2.2.2.3. Content Authors

2.2.2.2.3.1. TBD
3. Design Narrative

3.1. Synopsis

3.1.1. This design is for the implementation of the Campus WCMS Service (WCMS). The WCMS is an enterprise, campus-wide, mission-critical service that provides easy web content editing via the Cascade Server application and a new highly-available (HA) web layer to serve content. The Cascade Server application requires a SQL DB, of which we will employ 64-bit MySQL v5.1.

3.1.2. All local environment tiers will be utilizing existing Virtual Machine (VM) technology and hosted in the UCSC Datacenter. The VM cluster provides higher availability than a cluster of physical servers by abstracting the host layer to the entire cluster, but it is not completely immune to outages.

3.1.3. There will be three primary local environments (Prod, Stage and Dev)

3.1.3.1. The production (Prod) environment is the live business environment and the only environment used by clients (web authors, developers) of the WCMS service.

3.1.3.2. The staging (Stage) environment will be identical to production and is intended as an environment to test changes to production. The cascade staging environment will not be accessible by WCMS clients. The web staging environment will be used for pre-production ("pre-flight") web content created on the production cascade layer and will need to be visible to clients.

3.1.3.3. The dev (Dev) environment is designed primarily for the application administrators to investigate issues and deploy new features. Application administrators will routinely require full elevated privileges ("full sudo") on the dev environment.

3.1.4. There will be a Disaster Recovery (DR) environment in a remote location (TBD).

3.1.4.1. It is recommended that only the web layer be replicated in the DR environment for efficiency and resource concerns. How deep to replicate web content beyond www.ucsc.edu is TBD.

3.1.5. There will be a separate "training" instance of Cascade Server on the Stage environment. This will require a separate DB.

3.1.6. There may be a need for a media server for storing large files outside of the MySQL database. Currently the need for this is unknown. The "cold" cascade server in the production environment may be used for this purpose in a limited fashion until more is known.

3.2. Controls

3.2.1. All changes to production will follow the ITS change management process by deploying first in Dev, verified in Stage when necessary and finally implemented on Prod.

3.2.2. A Security Scan will be performed by IT Security against prod, stage and dev pre-deployment and at regular bi-annual intervals.

3.3. Current Inventory

3.3.1. There are no plans to reuse or deprecate any currently running server or service when this service is deployed. Over time, the "www" web environment may be more likely to be deprecated once all the (25) sites and applications (News & Events and Campus Calendar) on that system are migrated to the WCMS.

3.4. Classification

3.4.1. IS-2: There is no confidential or restricted data in any of the WCMS tiers or environments

3.4.2. IS-12: BC/DR options are TBD

3.5. Other Policies
3.5.1. Regarding Minimum Network Connectivity Requirements (MNCR): Although the WCMS servers are not intended to store confidential or restricted data, there is no way to prevent users from entering sensitive data.

3.5.2.IS-11: The Cascade Server application will be using Shibboleth for authentication and no passwords will be transmitted or stored locally.

4. Solution Diagram
4.1. High Level Diagram
4.2. See Appendix C - TBD

5. Technical Schematic
5.1. See Appendix D
5.1.1. Physical Diagram
5.1.2. Logical Diagram
5.2. Vendor Requirements - http://www.hannonhill.com/products/requirements.html

- Client Machine Requirements (for content contributors)
  - Compatible Browser
    - Internet Explorer 6 or higher on Windows
    - Firefox 2/3 on Mac OS X, Windows, and Linux
    - Safari 3/4 on Mac OS X, Windows
  - Display capable of 1024 x 768 resolution or higher
  - Network connection

- Server Requirements (for CMS deployment)
  - Windows 2000/XP/2003/Vista, Linux, Solaris, Mac OS X operating system
  - MySQL 4.1/5.0/5.1, MS SQL Server 2005, Oracle 9i/10g
  - Java 5.0 (1.5) JRE on CMS server machine
  - Apache, Microsoft IIS, Netscape Enterprise Server, Zeus Web Servers
  - Minimum Dual Core Pentium /AMD Athlon or equivalent
  - Minimum 3 GB RAM available
  - Minimum 1 GB HD space available

6. Application
6.1. Synopsis - WCMS uses the Hannon Hill (vendor) Cascade Server (product), deployed in an n-tier model – including layers for Web, App, and Database. There will be additional layers for Media, Utilities, and Training. There will be Dev, Stage, and Production environments established. All applications will reside in /opt/app directory on their respective hosts unless otherwise noted.

6.1.1. Apache web server (on the web layer)
   6.1.1.1. This layer receives static HTML files and entire websites published via SFTP from the Cascade server.
   6.1.1.2. All HTML files will be located in /opt/app/root/apache/htdocs/[CNAME] (TBD)
   6.1.1.3. Each website will execute as a separate user/thread (utilizing suEXEC & suPHP)
   6.1.1.4. This layer will not serve non-Cascade published websites
   6.1.1.5. All dynamic and/or “pass-thru” code must meet the following (draft) requirements, in order to contain runaway processes and other risks coming from server-side code:
     6.1.1.5.1. All code must be stored in source-control
6.1.1.5.2. All code must be tested first on another environment
6.1.1.5.3. All code must pass peer code review
6.1.1.5.4. All code must be written by ITS or “certified” developer
6.1.1.6. Only app admins will be able to login to the web layer; users and developers will not have access to the web layer
6.1.1.6.1. Note: exceptions to this will be granted for emergency-only situations
6.1.1.7. Load-balanced configuration
6.1.1.7.1. The two web layer VMs will be behind 2 network load-balancer appliances
6.1.1.7.2. A script will be written to verify web content is identical on both web servers (specifics TBD)
6.1.1.8. Technical configuration
6.1.1.8.1. Apache v2.2
6.1.1.8.1.1. suEXEC and suPHP modules
6.1.1.8.2. PHP 5.1.6
6.1.1.8.3. OpenSSL
6.1.1.9. Only ports 80 (http) and 443 (https) will be open to the public
6.1.1.10. Port 22 (ssh & sftp) will be open only from the datacenter

6.1.2. Cascade Server & Tomcat (on the app layer)
6.1.2.1. Cascade version 6.x
6.1.2.2. Tomcat comes bundled with Cascade; No identified need to install separately
6.1.2.3. We will deploy apache as a proxy server (using mod_proxy or mod_ajp) in front of tomcat and cascade; This will allow the ability to utilize shibboleth, better logging (via apache logs), the ability to quickly post a maintenance page for the cascade server and better performance.
6.1.2.4. We may also run the Java HotSpot Server VM instead of the JRE that comes with the cascade installer for increased performance. Reference College William & Mary
6.1.2.5. Preferred architecture is having two licensed production cascade servers.
6.1.2.5.1. A purchasing decision was made to only license a single production cascade server. This adds availability risk to the production environment and adds additional management and client overhead to schedule normal and out-of-cycle maintenance to the production cascade server.
6.1.2.5.2. A “cold” standby cascade server will be setup in the production environment in case of failure of the “hot” production cascade server.
6.1.2.5.3. Complexity, duration and reliability of syncing and failing over from hot to cold cascade server unknown and TBD
6.1.2.6. Shibboleth authentication for Cascade Server will be utilized using Apache mod_shib and SSL
6.1.2.6.1. There will be no network restrictions for cascade server login
6.1.2.7. Port 22 (ssh) will be open only from the datacenter networks
6.1.2.7.1. Only systems admins and app admins will be able to SSH into this host; unless emergency access granted
6.1.2.8. Port 3306 (mysql) will only allow traffic to/from the MySQL DB VIP
6.1.2.9. Maintenance Windows for Cascade Server - Proposed
6.1.2.9.1. Weekly: Thursdays 7-11pm
6.1.2.9.2. Monthly: 1ST Saturday/month 8am -5pm
6.1.2.10. Cascade Server account mgmt
6.1.2.10.1. TBD
6.1.2.11. Cascade Server application configuration
6.1.2.11.1. TBD

6.1.3. MySQL (on the DB layer)
6.1.3.1. See WCMS Database Architecture document for detailed discussion
6.1.3.2. MySQL will be v5.1 64-bit (InnoDB) on Solaris x86 in VM using multi-master manager (MMM) scripts for high availability
6.1.3.3. For production, MySQL will be deployed in a multi-master configuration using a virtual IP address for the cascade server application as the client. This setup allows for immediate and
automatic failover to a secondary DB server in case the master server becomes offline or needs disruptive attention (planned maintenance).

6.1.3.4. Switching from the master to the slave DB is done automatically by the MMM scripts, or manually by the DBA team.

6.1.4. Media Server

6.1.4.1. We may use the cold cascade server as a temporary media server.

6.1.4.2. Another option would be to use AFS for a temporary media server. This would utilize the existing SFTP server (sftp.ic.ucsc.edu), web server (www2.ucsc.edu), and account/file mgmt (Moira).

6.1.4.3. Details (quota, access, etc) TBD

6.1.5. Web Utility Server

6.1.5.1. This is a separate standalone server located in the staging environment.

6.1.5.2. The following utilities will be installed in order to better manage ITS Web Services:

6.1.5.2.1. Tripwire
6.1.5.2.2. Monit/BigSister/Nagios
6.1.5.2.3. Web application security scanning (TBD)
6.1.5.2.4. SVN
6.1.5.2.5. Website & application logging and analysis
6.1.5.2.6. other scripts/libraries as needed

6.1.6. Training instance

6.1.6.1. A separate instance of Cascade Server will be setup in the staging environment for user training, most likely on a separate port.

6.1.6.2. A “training” app and DB configuration snapshot will be utilized to reset training environment.

6.1.7. “Pre-flight” web publishing

6.1.7.1. Web authors may want a pre-production web environment to publish partial pages or their full website before publishing to production. For this, we will setup a “pre-flight” publish function from the production cascade instance to the Stage web servers.

6.1.7.2. As passthru/dynamic code will at times be published via cascade, publishing preflight pages to the staging environment is preferred to the production environment for availability and security precautions.

6.1.7.3. Websites will be named http://cname-preflight.ucsc.edu

6.1.7.4. Note: Hannon Hill may provide input on best practices with this.

6.1.8. STAGE and DEV environments

6.1.8.1. The staging environment is primarily designed for testing changes on a prod-equivalent environment. Priority will be given to testing changes over the training and pre-flight instances.

6.1.8.2. Except for the training and pre-flight instances, Stage and Dev will only be accessible via the 128.114.x.x UCSC network

6.2. Compatibility/Interoperability

6.2.1. Cascade Server is accessible from all supported browsers (Firefox, Safari, IE) on all supported operating systems (Windows, Mac, Linux)

7. Data

7.1. Data Model – N/A

7.2. Data Migration - Content will be manually migrated from existing web systems (www and www2). Migration design details are TBD.

8. Interface Definition

8.1. AIS course data import - TBD

8.2. Campus Calendar - TBD
9. **Add’l Components/Functionality**

   9.1. Utilities host

   9.1.1. Currently ITS Web Services does not have a host to perform req’d scanning, monitoring and change control

   9.2. Media Server

   9.2.1. Cascade natively stores all files in the DB. This is not optimum for storing large blob files such as images or PDFs. Many deployments have setup a separate media server to handle this. Since we don’t currently know this demand, I propose to use the “cold” cascade server in production as a temporary media server. If a stand-alone media server is warranted, we can build that out later

   9.3. SVN

   9.3.1. Where possible, application configuration files will be stored in SVN repos.

   9.3.2. Note, the SVN service is informal and adhoc.

10. **Security**

    10.1. See Appendix A for MNCR results

    10.2. See Appendix B for IS-3 results

    10.3. There is no PII, PCI, HIPAA or other confidential or restricted data in the system

    10.4. FERPA registered students may be granted access to log into Cascade. FERPA policies will be adhered to where necessary.

    10.5. All Systems are located in the Datacenter which requires Staff ID and omnilock codes to enter

    10.6. All Systems will be behind DC firewalls and also have HBF installed (iptables)

    10.7. Cascade requires login; login will be shibbolized

    10.8. Content from cascade is transmitted via SFTP to the web layer server(s)

    10.9. See Appendix E for Security Issue Matrix

11. **Service Model**

    11.1. User guides/FAQs will be on the web layer

    11.2. Maintenance Windows: See Item 6.1.2.9

    11.3. Change Management utilized for release procedures

    11.4. Technicians required depend on release specifics

    11.5. Standard Backup and Recovery services done via Core Tech.
Appendix A

Minimum Network Connectivity Requirements

A. Access Control Measures
All logins to cascade are password protected

B. Encrypted Authentication
User authentication will be done via Shibboleth (and SSL)

C. Patch Management Practices
Monthly scheduled patches will be tested on the dev environment prior to application to production

D. Malicious Software Protection
All systems will run said applications

E. Removal of Unnecessary Services
All systems will run only necessary services/daemons

F. Host-based Firewall Software
HBF will be run on all servers

G. Authenticated Email Relay
Servers will not run an email relay

H. Authenticated Network Proxy Servers
Servers will not run proxy servers

I. Session Timeout - Session Timeout will be set to 60 minutes

Appendix B - IS-3 results
(See Excel document)

Appendix C - Solution Diagram
TBD

Appendix D - Technical Schematic
(see Visio diagram)
## Appendix E - Security Issue Matrix

| Security vulnerability in RHEL5, MySQL, Apache, PHP, Cascade, Tomcat, SSL, SSH | High | High | High | 1. Monthly patching  
2.  
3.  | 1. L/L  
2.  
3.  | 1. Regular maintenance windows  
2. PACs for applying patches |
| --- | --- | --- | --- | --- | --- | --- |

**Summary: [Is residual risk accepted?]**

| Vulnerability introduced in dynamic/passthru code | Med | High | Low | 1. Code review prior to publish  
2. Regular vulnerability scanning  
3.  
4. | 1. H/L  
2. M/M  
3. | 1. Develop code review practice  
2. Determine and schedule app scanner |
| --- | --- | --- | --- | --- | --- | --- |

**Summary: [Is residual risk accepted?]**

| Compromise of cascade administrator password | Low | High | High | 1. Limit distribution of admin password  
2. Change admin password qtrly  
3. | 1. L/L  
2. M/L  
3. | 1. Just do it |
| --- | --- | --- | --- | --- | --- | --- |

**Summary: [Is residual risk accepted?]**
## Current Open Issues (as of 1/6/10):

<table>
<thead>
<tr>
<th>#</th>
<th>Owner</th>
<th>Status (Open, In Progress, Closed)</th>
<th>Priority</th>
<th>Issue</th>
<th>Comments/Solution/CollabIssue#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Andre</td>
<td>In Progress</td>
<td>Med</td>
<td>Cascade/Auth: Shibboleth sign out issue → ForceAuth</td>
<td>#16 (Use ForceAuth)</td>
</tr>
<tr>
<td>2</td>
<td>Andre</td>
<td>Closed</td>
<td>Low</td>
<td>HH/Cascade/Ops: Use Apache proxy in front of Tomcat for cascade? Mod_ajp?</td>
<td>#17 (mod_ajp or mod_proxy both work fine - as per HH)</td>
</tr>
<tr>
<td>3</td>
<td>Andre</td>
<td>Closed</td>
<td>Low</td>
<td>HH/Cascade/Perfil: Use Java Hotspot Server JVM to increase cascade performance</td>
<td>#18 (built-in JVM and JHS both work fine – as per HH)</td>
</tr>
<tr>
<td>4</td>
<td>In Progress</td>
<td>High</td>
<td></td>
<td>HH/DB/Perf: Large files in DB; use Media server? → Media server = cascade cold server?</td>
<td>#9 (Can set quota on uploaded files in Cascade configuration; Will work with DBA group to establish quota levels; Will design prototype media server)</td>
</tr>
<tr>
<td>5</td>
<td>Charles,</td>
<td>In Progress</td>
<td>High</td>
<td>Service: better understanding of sysadmin/appadmin roles &amp; responsibilities re: patching, incident response, privileges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Charles,</td>
<td>Open</td>
<td>High</td>
<td>Security/Ops: Insure OS builds are minimal, no add’l services are running</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Charles,</td>
<td>Open</td>
<td>High</td>
<td>HH/Security/Ops: Controlling server code on web layer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>David</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Andre,</td>
<td>Open</td>
<td>Med</td>
<td>Ops: Tripwire for cascade? → utility server</td>
<td>Install on utility server. Potential conflict with core tripwire server?</td>
</tr>
<tr>
<td></td>
<td>Jacques</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Andre</td>
<td>Open</td>
<td>Low</td>
<td>HH/Ops: Performance implications with one production cascade server vs two</td>
<td>TBD</td>
</tr>
<tr>
<td>10</td>
<td>Andre</td>
<td>In Progress</td>
<td>High</td>
<td>HH/Ops: Designing &amp; executing failover to “cold” cascade server</td>
<td>#19</td>
</tr>
<tr>
<td>11</td>
<td>Raji</td>
<td>In Progress</td>
<td>High</td>
<td>HH/Ops: Designing &amp; implementing HA DB layer</td>
<td>#20 (Raj to provide MMM design by 12/18)</td>
</tr>
<tr>
<td>12</td>
<td>Charles</td>
<td>In Progress</td>
<td>Med</td>
<td>Ops: Support levels for App/DB layer</td>
<td>Application Layer: 7am – 7pm, 4 hour recovery</td>
</tr>
<tr>
<td>13</td>
<td>Jacques,</td>
<td>In Progress</td>
<td>Med</td>
<td>Ops/Perf: Performance requirements for web layer?</td>
<td>(same as current environments)</td>
</tr>
<tr>
<td></td>
<td>Charles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Raji</td>
<td>Closed</td>
<td>High</td>
<td>Ops: RHEL5 or Solaris for DB layer?</td>
<td>Solaris x86 in VM</td>
</tr>
<tr>
<td>15</td>
<td>Charles</td>
<td>Closed</td>
<td>High</td>
<td>HH/Ops: Version of Cascade for production (6.2.1 or 6.4)?</td>
<td>Sandstorm = 6.4.0.1</td>
</tr>
<tr>
<td>16</td>
<td>Charles</td>
<td>Open</td>
<td>Med</td>
<td>HH/DR: DR design (web layer only or full stack)?</td>
<td>Recommend limited web layer only</td>
</tr>
<tr>
<td>17</td>
<td>Charles</td>
<td>Open</td>
<td>Med</td>
<td>DR: How to provide direct access to web content during DR event</td>
<td>#21</td>
</tr>
<tr>
<td>18</td>
<td>Charles,</td>
<td>Open</td>
<td>Med</td>
<td>DR: DNS magic...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>John H</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Charles</td>
<td>Open</td>
<td>Med</td>
<td>DR: where to host?</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Andre</td>
<td>Open</td>
<td>Med</td>
<td>HH/Ops: Admin password mgmt</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Charles</td>
<td>Open</td>
<td>Low</td>
<td>Build: Performance metrics/expectations (PIO, Hum, Porter)</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Charles</td>
<td>Closed</td>
<td>Med</td>
<td>HH/Do we need 2 staging web servers?</td>
<td>Yes</td>
</tr>
<tr>
<td>23</td>
<td>Andre,</td>
<td>Closed</td>
<td>Med</td>
<td>HH/Authentication if Shibboleth fails?</td>
<td>Assign some admin accounts &quot;Normal&quot; (aka local) authentication</td>
</tr>
<tr>
<td></td>
<td>Eric G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Charles</td>
<td>In Progress</td>
<td>Med</td>
<td>HH: Process for upgrading minor/major CS releases?</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Andre</td>
<td>Open</td>
<td>Med</td>
<td>Cascade Config files in SVN?</td>
<td>#22</td>
</tr>
<tr>
<td>27</td>
<td>Andre</td>
<td>Open</td>
<td>Med</td>
<td>Build “training” instance on stage</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Jacques</td>
<td>Open</td>
<td>High</td>
<td>Best practice for sync-ing content on multiple web servers</td>
<td>#23</td>
</tr>
<tr>
<td>29</td>
<td>Charles</td>
<td>In Progress</td>
<td>Med</td>
<td>Design for “staging” publishing environment in prod</td>
<td>Use 2&quot;nd layer staging web layer</td>
</tr>
</tbody>
</table>
**Current Specs of www & www2 production environments**

<table>
<thead>
<tr>
<th></th>
<th>Site count</th>
<th>Pages</th>
<th>DBs</th>
<th>HDD usage</th>
<th>Hits/Sec AVG:MAX</th>
<th>KB/Sec AVG:MAX</th>
<th># servers</th>
<th>CPU</th>
<th>CPU usage (Avg:Max)</th>
<th>RAM</th>
<th>RAM usage (Avg:Max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>www</td>
<td>26</td>
<td>43K</td>
<td>2</td>
<td>20GB</td>
<td>6.5:9.0</td>
<td>n/a</td>
<td>2 VM</td>
<td>Windows/IIS</td>
<td>3Ghz (single-core)</td>
<td>3%:70%</td>
<td>1GB</td>
</tr>
<tr>
<td>www2</td>
<td>240</td>
<td>1.5M (prod &amp; dev)</td>
<td>33</td>
<td>62GB (prod &amp; dev)</td>
<td>.4:9</td>
<td>34:114</td>
<td>2 physical</td>
<td>RHELS/Apache</td>
<td>4 Xeon 2.33Ghz</td>
<td>3%:10%</td>
<td>4GB</td>
</tr>
</tbody>
</table>

**Technical Specifications**

**Production**

<table>
<thead>
<tr>
<th>Comments</th>
<th>Server</th>
<th>OS</th>
<th>Apps</th>
<th>CPU</th>
<th>RAM</th>
<th>HDD</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>2 VMs (network load-balanced)</td>
<td>RHEL5</td>
<td>Apache</td>
<td>2 dual-core</td>
<td>4GB</td>
<td>200GB</td>
<td>24/7</td>
</tr>
<tr>
<td>App</td>
<td>2 VMs (1 hot; 1 cold standby)</td>
<td>RHEL5</td>
<td>Cascade 6.x Tomcat Java 5</td>
<td>2 dual-core</td>
<td>8GB</td>
<td>100GB HDD 250GB HDD</td>
<td>7-7</td>
</tr>
<tr>
<td>DB</td>
<td>3 VMs (2 masters; 1 manager)</td>
<td>Solaris10 x86</td>
<td>MySQL 5.1 64bit InnoDB</td>
<td>1 dual-core</td>
<td>8GB</td>
<td>200GB</td>
<td>7-7</td>
</tr>
</tbody>
</table>

**Stage** – same as Production except for the following:

1. Restrict traffic to on-campus only (128.114.x.x) – for ITS use only
2. Allow network traffic from outside campus for training and preflight instances only
3. Web: 8-5 support
4. App: Same as prod except no cold standby (one 1 VM)
5. DB: use separate DB on production server suite