Technical REport Draft

Real-Time Score Board Mid-Semester interview assessment

Abstract

To inform and advise the client and future teams undertaking this project, if this is the case, we have addressed the technical needs and recommend solutions for the project under discussion.

There are 3 main factors to be presented which involves analyses of the venue Infrastructure and current Hardware, hardware solution evaluation, cloud based solution evaluation. At last a recommendation is given based on the information presented on each section, it aims to address the best resolution to the problem domain.

The proposed solution for a software was a web-based application which is within our believes the best match for the client requirements. However, our assessment is not limited to one approach.

Venue Infrastructure

Hardware requirements

This section aims to evaluate the hardware necessary to host the application in a local environment. As many options are available to be chosen from our goal is to recommend the adequate hardware to satisfy the client's need.

It is important to highlight that this research was undertaken as an individual component of the project and it disregards any information about the venue current infrastructure such as networking and or possible 3rd part solutions such as cloud based. A final recommendation will be provided in the recommendation section.

Any pricing provided was acquired through the research process, all prices a subject to changes.

The research was done based on the requirements of the system, see below:

* Project goal is to create a **real-time** scoreboard application\website.
* Can handle 100 users(judges) updating the database simultaneously.
* Can handle at least 400 users() or requests simultaneously.
* OS was not specified as client required a open source "system". AUT could also aid with software.

**1st recommendation**

In order to host a back-end system for either a web-based-application or for a mobile phone application We recommend the system to be divided in three servers: Reverse Proxy, Application Server and Database server, please see figure 1 for a high level network draft. For reliability and performance, it is recommended that you separate them together. (*See distributed system reference for more info*).

**Please see appendix A for hardware specifications and quotation.**

Machine generated alternative text:
RESTful request 
Internet/ 
Response with ISON 
Application 
Figure 1 : 
Senaer Side 
Reverse Proxy server (NGINX Server) 
Application server 
(Tomcat server) 
Database server (Postgresql) 
Discussed framework 

**2nd solution**

A more compact solution that would also impact on the cost of the system is to take use of virtualization (*See virtualization in reference for more info*). The use of this strategy would allow

to co-populate 2 servers on one host, that is one server would have role of two, in this case the application server and the reverse proxy server will be in one server. This strategy would eliminate the need of one server. However, some enhancements will be need such as more memory RAM capacity.

Apart of the cost benefits, using virtualization take better use of the processing power of a server. Taking in consideration that most of the requests will be **I/O bound** other than **CPU bond**, the hardware recommended would not have problem responding to the demand.( *See Request-per-second in reference* )

However, there is a higher reliability on the server in discussion and therefore the risks are also higher. For the same reason, we recommend to have the Database server as a separate system, in simple words, there is where all your data will be stored.

**Please see appendix A for hardware specifications and quotation.**

**3rd Solution**

To have an application that supports images and stores data, a database and a server should be enough and many times a proxy is not used. For light traffic it could even be a machine that provides both services.

The traffic expected for this application is consider, in our opinion, medium to high traffic allowing space for scalability. So for reliability purposes we would recommend two independent serves to decrease the risk of failure and data loss.

This set up could provide a working environment to the application but it would still face potential issues such slowness or no response, security issues, and possible crash of the whole server\system.

The use of a reverse proxy server would among many other benefits distribute the load from incoming requests, protect against common web-based attacks, like DoS or DDoS and malware and reduce load on its origin servers by caching static content.

Therefore, we discourage this solution as there many components which could lead to project failure.

**Disaster management**

What happens if there is a sudden power outage?

Overheating of the servers CAN cause fire, what would you do if anything as such happens?

**Maintenance**

Cost to run the equipment’s.

**Durability**

**Appendix A**

**Requirements:**

* Project goal is to create a **real-time** scoreboard application\website.
* Can handle 100 users(judges) updating the database simultaneously.
* Can handle at least 400 users() or requests simultaneously.
* OS was not specified as client required a open source "system". AUT could also aid with software.

**Assumptions:**

Database

1. HDD - 2 TB or more SSD enterprise grade in RAID set-up.

2. Processor - Intel Xenon processor that support Error Correct Code (ECC).

3. 128 Gb (minimum) with ECC.

4. Fail-over power for your server.

Reverse Proxy Server

1. HDD - 1 TB or more SSD enterprise grade in RAID set-up.

2. Two or more processors - Intel Xenon processor that support Error Correct Code (ECC).

3. 128 Gb (minimum) with ECC.

4. Fail-over power for your server.

Application Server

1. HDD - 1 TB or more SSD enterprise grade in RAID set-up.

2. Two or more processors - Intel Xenon processor that support Error Correct Code (ECC).

3. 128 Gb (minimum) with ECC.

4. Fail-over power for your server.

**Quotation**

Note: The computer specifications bellow are not only a guide. We believe that these are more than sufficient to comport the requirements above. Please see recommendation section.

Database Server

Hardware RAID controller with 2GB flash backed write cache

2x 150GB SSD (RAID1 – OS/Boot only – 150GB Raw usable)

6x 960GB SSD (RAID10 – Database – 2.8TB Raw usable)

Write workload max: 3.6TB per day

Reverse Proxy Server

Hardware RAID controller with 2GB flash backed write cache

2x 150GB SSD (RAID1 – OS/Boot only – 150GB Raw usable)

6x 480GB SSD (RAID10 – Hot Data – 1.4TB Raw usable)

Write workload max: 1.8TB per day

Application Server

Hardware RAID controller with 2GB flash backed write cache

2x 150GB SSD (RAID1 – OS/Boot only – 150GB Raw usable)

6x 480GB SSD (RAID10 – Hot Data – 1.4TB Raw usable)

Write workload max: 1.8TB per day

*\*\*\*Quotation has been attached separately.*

**References**

**Image**

Figure 1 - Retrieved from: Feedback 1. Author: Akshay Raj Gollahalli . 10th April,2017.

**Links**

Distributed Systems - <https://en.wikipedia.org/wiki/Distributed_computing> and <https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture>

Virtualization - <https://en.wikipedia.org/wiki/Virtualization>

Request per second - <https://wrongsideofmemphis.wordpress.com/2013/10/21/requests-per-second-a-reference/>