Hardware requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Author | Description | Reason | Version |
|  |  |  |  |  |
| 23/05/2017 | Vinicius | Initialization of report | Draft to be presented as a state report | 1.0 |
| 23/07/2017 |  | Update information\revision | Further explanation provided for each solution | 1.1 |

This section aims to evaluate the hardware necessary in order to host the application in a local environment. As many option 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.

Any pricing provided was acquired through the research process, all prices are 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 an open source "system". AUT could also provide assistance with software.

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

**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. (*See distributed system reference for more info*).

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.

The application server would be responsible for handling requests from user or proxy server. It would inquiry for information from database and serve it back to users as well as do any processing required by in each request.

The database is responsible for managing all data of the application and serve it back to the application server when requested.

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 located in one server. This strategy would eliminate the need of one server. However, some enhancements could be needed 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.

**3rd Solution**

A simpler approach is to have one sever that works as a database and application server and eliminate the need for a proxy server, usually for small applications or businesses this approach is considered.

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

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