# The Name of the Title is Hope

Ben Trovato*
G.K.M. Tobin*
trovato@corporation.com
webmaster@marysville-ohio.com
Institute for Clarity in Documentation
Dublin, Ohio

Lars Thørväld
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Valerie Béranger
Inria Paris-Rocquencourt
Rocquencourt, France

Aparna Patel
Rajiv Gandhi University
Doimukh, Arunachal Pradesh, India

Huifen Chan
Tsinghua University
Haidian Qu, Beijing Shi, China

Charles Palmer
Palmer Research Laboratories
San Antonio, Texas
cpalmer@prl.com

John Smith
The Thørväld Group
jsmith@affiliation.org

Julius P. Kumquat
The Kumquat Consortium
jpkumquat@consortium.net

## ABSTRACT

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

## KEYWORDS

datasets, neural networks, gaze detection, text tagging

## 1 INTRODUCTION

## 2 BACKGROUND

### 2.1 College Application Process

The automated college admission system is providing the SSC (Secondary School Certifacte) passed students a user friendly platform to apply for their desired colleges. The current automated system is serving from the academic year 2007-18 ?? and almost 1.3 millions of students submit their college choices under this application system. Proir to the application process, the applying student need to make the payment according to the general guideline published by Bangladesh Inter-education Board Coordination Subcommittee. For

*Both authors contributed equally to this research.

**Unpublished working draft. Not for distribution.**

example, the gudeline of the year 2020 directs that the applicants need to pay BDT150 through one of the registered Payment Service Providers (PSP); i) Nagad [4], ii) Sonali bank [1], iii) Teletalk [6], iv) bKash [3], v) Rocket [2]. As soon as the payment is confirmed, an applicant can start his/her application process by clicking the Apply Now button on the landing page of the web application of XI Admission System [5] (shown in Figure 1a). The button redirects the applicant to one of its existing application servers. This allocation is done using a DNS based load balancer that performs a modulus arithmetic on the timestamp of clicking the Apply Now button. All the services to the applicant is served from this specific application server. The applicant then need to get into the main application process by inputing his/her SSC Roll No, Board, Passong Year and Registration No as shown in Figure 1b. After logging into into the system, the applying student need to verify the contact no he have submitted during the payment procedure (Figure 1c). A successfull verification brings the applicant to the college choice page as shown in the Figure 1d. Each college of admission system is identified by an id named *EIIN*. Each college can have:

- multiple shifts such as Morning, Day, etc. which can be idenfied by *Shift ID*
- two possible versions such as English, Bangla which can be identified by *Version ID*
- multiple groups such as Science, Humanities, Business Studies, etc. which can be identified by *Group ID*

Now each college choice is a unique tuple of (EIIN, Shift ID, Version ID, Group ID) and can be termed as an ESVG choice. An applicant can add an arbitrary number of ESVG choice while keeping the count of unique EIIN within a range of minimum of five and a maximum of ten.

### 2.2 Load Testing

### 2.3 Load Testing Tools

## 3 ENVIRONMENT SETUP

We conduct our load testing on three different environment; i) on the test bed, ii) on the production environment in projected load

(a) Application Landing Page

(b) Student Log In Page

(c) Contact Number Validation Page

(d) Application College Choice Page

**Figure 1: Screenshots of Web Application of XI Class Admission System**

scneario, and iii) on the live application website in real scenario. The test environment is initially set up on a local server at the IICT. Testing on the test bed allows the authors to prepare a proper test plan and a benchmark to evaluate the performance of the final environment.

## 3.1 Server Configuration

*3.1.1 Test Environment.* The test environment is comprised with a single application server and a single database server running on Oralce 12c ? [?cite?]. The configuration of the test bed is as follows.

- Application server has 16 GB of RAM, 1 CPU, and 1 tomcat server running with 8 GB allocated memory.
- Database (DB) server has 24 GB of RAM and 1 CPU.
- Maximum allowed connections on the oracle DBMS
  - Session 252
  - Process 150
  - Open Cursor 300
- Maximum 30 connections in the connection pool from each tomcat to database.

*3.1.2 Production Environment.* As the time of main application reaches nearer, the college admission application system is deployed on a production environment with higher configuration setup. The production environment is built on the virtual machines hosted by the renowned cloud service provider named "Brilliant Cloud". This production environment is comprised of six application server and a single Oracle database server. All the tests from the test server are replicated on the production server and it is expected to get a significantly better performance from the test server. The The configuration of the production environment for College Admission 2020 is as follows.

- Each of the six Application servers has 32 GB of RAM, 8 core processor, and 100 GB Storage.
- Single Payment server has 32 GB of RAM, 8 core processor, and 100 GB of Storage.
- Single database server has 128 GB of RAM, 16 core processor, and 280 GB SSD Storage.
- Maximum allowed connections on the oracle DBMS
  - Session ~ 1.5 × Process

- – Process 2500
- – Open Cursor 4000
- – shared server 1,
- – hared pool = 0
- Provision of dedicated server which means 1 process is 1 dedicated connection.

## 3.2 Load Balancing on Production Environment

As we already have mentioned that the final environent is comosed of six appliation servers which are used for load balancing. The college admission applicaton system uses DNS based load balancing technique. DNS load balancing is a software-defined approach to load balancing where client requests to a domain within the Domain Name System (DNS) are distributed across different server machines. The DNS system sends a different version of the list of IP addresses each time it responds to a new client request using the round-robin method, therefore distributing the DNS requests evenly to different servers to handle the overall load. This in turn provides DNS load balancing failover protection through automatic removal of non-responsive servers. Each time a client is trying to hit the application page, the timestamp is recorded, and based on the modulus arithmatic on this timestamp, the client is directed to a specific server among the six to handle his process of application.

Figure 2 shows the system architecture of the college admission application system. Each pf six application server are tagged with xiapp2–xiapp7.
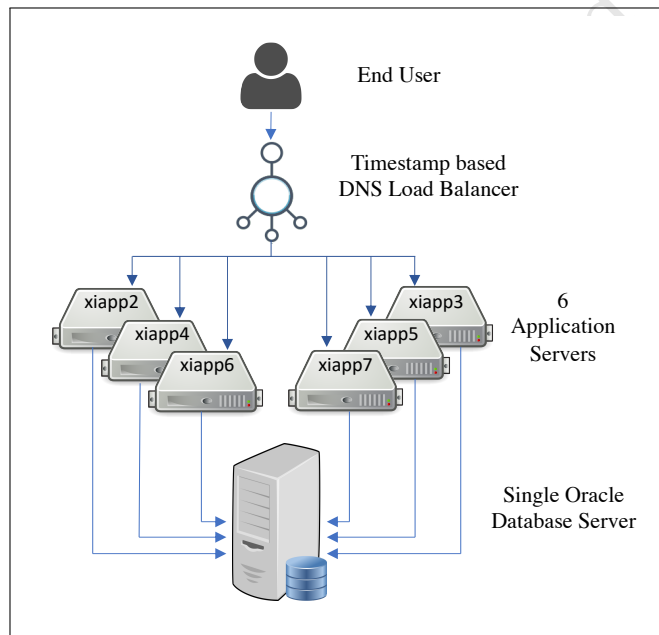


**Figure 2: Artchitecture of the Final Production Environemnt**

## 4 TEST PLAN

We prepare a suitable test plan for this year's application system based on the historical data of last years. The following historical

data is extracted from the college admission application process of 2019.

- Maximum hourly load was ~ 18684 applications, which happened at mid-day of day-1.
- 38 applications per second is a burst rate that happens in around mid-day.
- Applicant count was 1373387.
- No of SMS applications: 349986, No of WEB applications: 5741143.
- ESVG count 6091129. Therefore, average ESVG by a student was ~ 4.5.
- Maximum ESVG choices done by a student was ~ 28.

Based on the statistics from the previous year, a load projection plan is prepared as follows.

- We plan to load the system with a maximum hourly rate of 25000 application.
- The server performance should be examined by generating a varying rate of load between 5000 applications/hour and the maximum rate.
- Since there are database read and write operation involved in the application process, the impact of database size on the application response times is needed to be examined.
- We plan to observe that if a burst rate of at least 50 is generated under the projected load scenario.
- The performance shold be tested by generating a varied number of concurrent connections. The target is to exhaust the system with high number of simutaneous connection and observe that if it can sustain with acceptable performance.

We should keep in mind that the test server may not be able to meet all our desired performance. However, we are intended to have a clear visualization on how well the production server is going to perform in real load scenario.

## 5 EVALUATION METRICS

We use the following metrics to evaluate the response time of each of the HTTP requests as well as the full application procedure.

- Average
- Median
- $90^{th}$ Percentile
- $95^{th}$ Percentile
- $99^{th}$ Percentile

In addition, we also use the percentage of error as an evaluation metric to assess how successfully the requests are served by the application and database server.

## 6 EXPERIMENTAL OUTCOME

### 6.1 Varied Load Projection

As we have discussed earlier, we project a load of application processes which we vary from 5000 applications/hour to a maximum of 25000 applications/hour rate. This test is carried out on both the test bed and the prduction environment. Each rate is generated for multiple times and an average is considered. During this test the load was generated from a machine which is connceted to a network different from the network where the application and database server are hosted. The main intention of this test is to assess if

the application system can provide services in a consistent manner and reach the maximum rate of application without compromising in the performace. Figure 3 demonstrate the variation in response times with the variation in projected load. It can clearly be observed that the preformance of the production environment is more stable with a shorter range of variation.

## 6.2 Impact of Database Partitioning

## 6.3 Imapct of Database Size

Each application process is started with a database query which search the details information of the applicant with the Roll No, Board, Passing Year, and Registration No provided at the log in page of Figure 1b. On the other hand, this process is ended with a database with of the applicant's college choices. Therefore, we intend to observe the impact of database size on the response times of the application processes. To evaluate that, we submit more than 400 thousand full applications each of which submits 10 ESVG choices.

Note that we perform these experiments at a fixed load of 20,000 applications per hour. The machines which are used to generate load for both test bed and final environment are connected with the same corresponding networks where the application and database servers are running. However, for the final environment, since the network bandwidth provided by "Billiant Cloud" is very high, the response times are similar for machines outside the system's network.

Figure ?? shows the impact of database size on the response times based on the previously mentioned metrics. Each graph shows the response times of a full application process for both the test and production environment.

Graphs are drawn for top two time consuming HTTP requests, "Application Log In" and "Application Submission" and the time requirement of a full application process. The response time are considerably flat with respect to the size of the database. However there is a spike for database size of 50,000 which can be an outcome of randomly introduced network latency. On the other hand, Figure ?? shows the similar results when the tests are conducted on the final environment. One point is cleat that, for both cases the response times stay considerably flat with the variation of database size. However, for the final server, all the response times are increased by some amount. For example, the median response time of the HTTP request for "Application Submission" takes 60$ms$ in the test environment. In contrary, it is increased to around 140$ms$ in the final environment. This is because of the slower database write operation which is caused by a slower SSD storage media of the virtual machines provided by "Brilliant Cloud". However, since all the response times were consistent and within a tolerable limit of delay, the test results were accepted by the Quality Assurance Team.

## 6.4 Varied Concurrent Connections

We vary the number of concurrent connections for both the test environment and the final production environment and record the response times for each of the HTTP requests as well as the full application process. We demonstrate the outcoms graphically in Figure 5.

Firstly, Figure ?? and Figure ?? show the results graphically for the test and final environments repectively. The Thana Retrieval, College Retrieval, College SVG Retrieval lines are drawn by averaging within all the existing HTTP requests executing the corresponding retrieval tasks. We can observe that response time of each kind tends to increase with the increased number of concurrent connections. On the other hand, except from some spikes in the landing pages, the response times for the main requests stay flat which bears the testimony that the final server is capable for serving smoothly under high concurrent load condition.

Secondly, Figure ?? shows the error generated with the increment in concurrency level. We can observe that the error rate stays within very small margin (maximum 1.4% to be specific) for the final environment which is comparatively higher (up to 10%) in the test environment.

Finally, Figure ?? shows response times based on different kind of metrics for full application process with varied concurrency rate. In case of the test conducted on the test environment, we can observe that all the metrics is increasing with the incease in concurrency while for the final environment, the average and median response times are staying sufficiently flat in the curves that provides the verdict that the final environment is very capable of performing consistently under high concurrent connections.

From the experiment we perform by varying the number of concurrent connection, we can conclude that the application servers in both the test and final environments are successful to serve within a tolerable error margin with a decent performance even with a large number of concurrent connections. We should keep in mind that the test environment consists of a single server machine while the final production environment has six such server machines to serve in parallel. Therefore, it is very expected that the final environment is delivering significantly better performance.

## REFERENCES

[1] [n.d.]. Nagad Digital Financial Service; a venture by the Bangladesh Post Office. Public Limited Company with 100the government. Retrieved August 20, 2020 from http://sonalibank.com.bd
[2] Dutch-Bangla Bank. [n.d.]. bKash Limited (bKash). Mobile Banking. Retrieved August 20, 2020 from https://www.dutchbanglabank.com/rocket/rocket.html
[3] BRAC Bank Limited. [n.d.]. bKash Limited (bKash). Mobile Banking. Retrieved August 20, 2020 from https://www.bkash.com/
[4] Third Wave Technologies Limited. [n.d.]. Nagad Digital Financial Service; a venture by the Bangladesh Post Office. Website. Retrieved August 20, 2020 from https://nagad.com.bd
[5] Bangladesh Ministry of Education. 2020. XI CLASS ADMISSION SYSTEM (SESSION 2020 - 2021). Website. Retrieved August 20, 2020 from http://xiclassadmission.gov.bd
[6] Government of the Peoples Republic of Bangladesh. [n.d.]. Teletalk Bangladesh Limited. Leading Mobile Phone Operator. Retrieved August 20, 2020 from http://www.teletalk.com.bd/
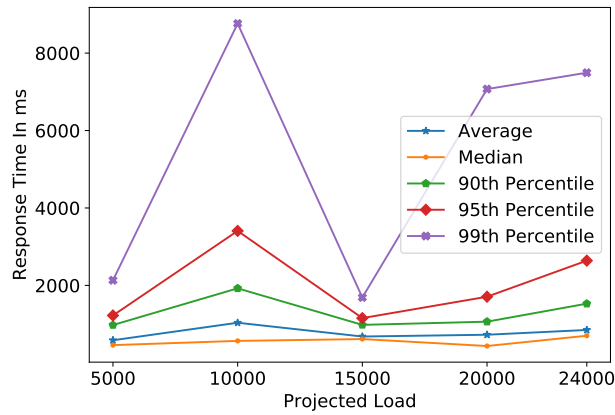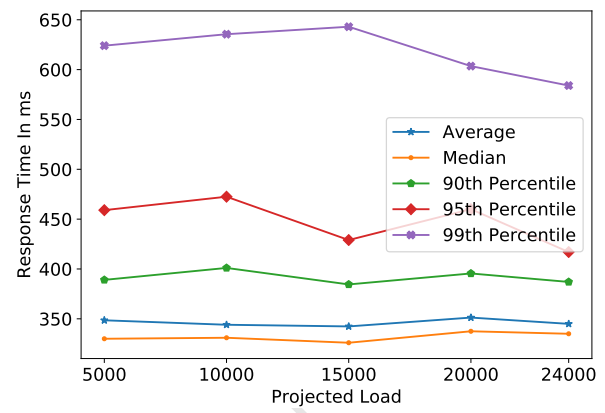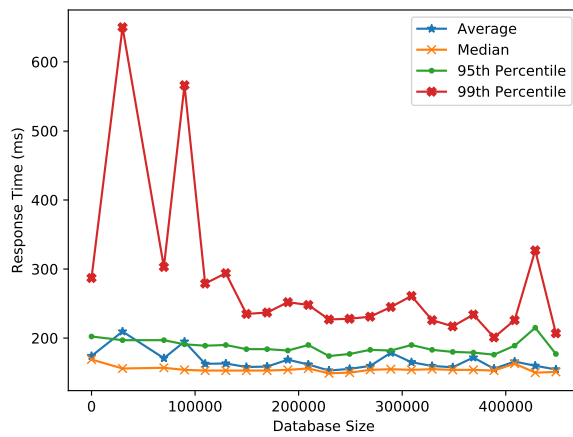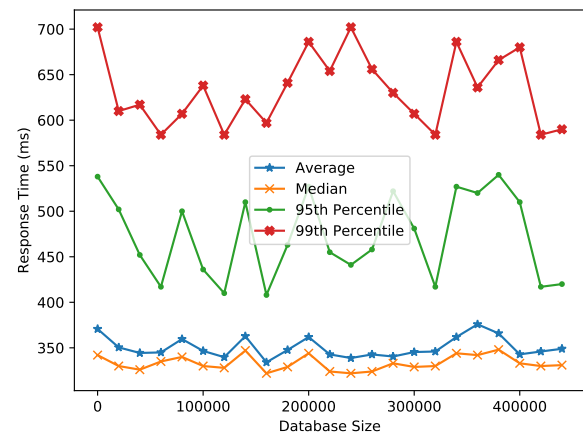
(a) Test Environment

(b) Production Environment

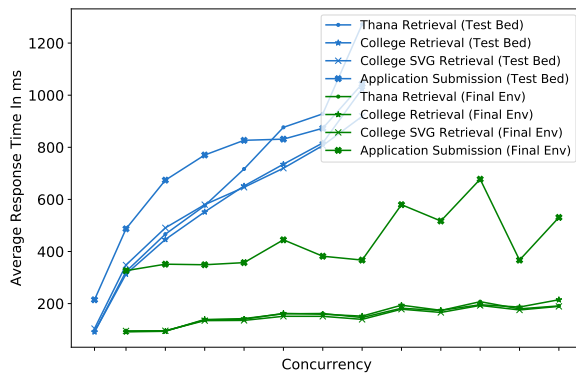Figure 3: Change in Response Times with Varied Load
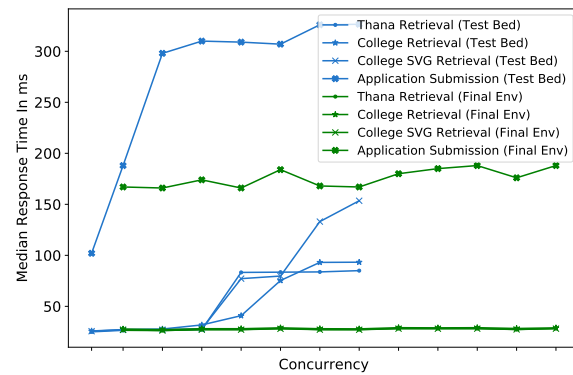


(a) Test Environment
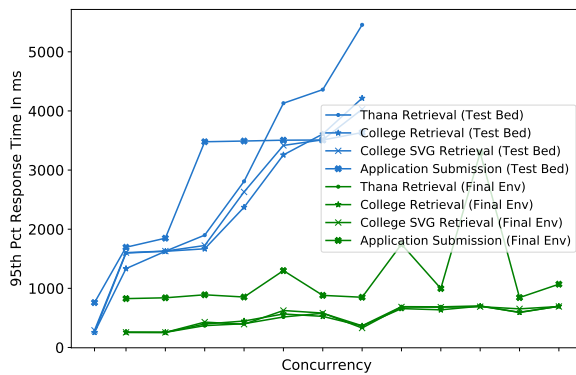
(b) Production Environment

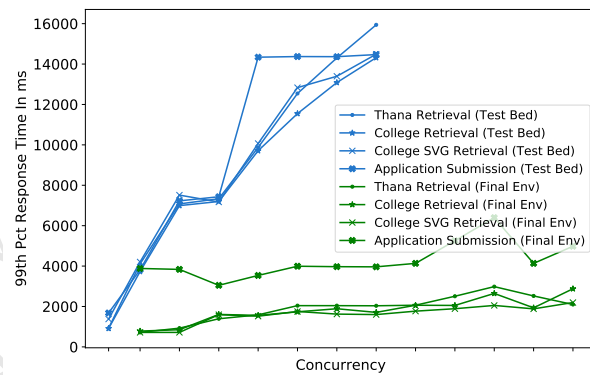Figure 4: Impact of Database Size on Response Times

(a)



(b)



(c)



(d)

**Figure 5: Impact of Varied Concurrent Connections on Response Times**