
Application:
World Hunger Foundation's *
On-Line Fundraising Site

Executive Summary

The World Hunger Foundation* commissioned Mentora Group, to perform an independent performance assessment of their On-Line Fundraising application to validate its ability to support an initial target load of 300 concurrent users.

The system tested is the WHF's production release hosted at AT&T's hosting facility in Ashburn, VA, running on Intel/Linux platforms, consisting of one web/application server, and a database server (table 2).

The simulated load applied to this environment was sized to simulate the peak sustained concurrent load anticipated during the projected highest usage four-hour evening period, when fundraisers are most likely to access the site.

Key findings (table1) quantify these measurements:

- **System Scalability** – The overall ability of the system to process transactions within a target range of acceptable time, at a target concurrent user load.
- **Page Presentation Response** – The ability of the system to display a page, within the target response time, after a user clicks the activating link.
- **Bottlenecks** – The system components whose resource utilizations indicate a system capacity limitation.
- **Bandwidth Utilization** – The ability of the system to support the total of “bytes sent and bytes received” of all objects on all pages included in the transactions tested, at the target load, with no measurable limitation imposed by the system.

We tested a mixed load of simulated users performing these three business transactions deemed to be the most numerous and resource intensive:

Table 1 – Key Findings

- System Scalability:** The system shows a gradual performance decline up to about 115 users, after which the target transaction completion time for Transaction B (4 minutes) is exceeded. **The system does not meet its load-response target of 300 users, reaching about 46% of this target load.** (See Table 3 for achieved transactions per hour).
- Page Presentation Response:** Average page presentation times stay within acceptable times (4 seconds max) up to about 115 users. **Above that number, this target is exceeded, degrading to about 8 seconds for the slowest transaction at the target load.** (See *Detailed Findings, Observation 2*).
- Bottlenecks:** **The key system bottleneck appears to be the database server, which maxes out at about 150 users.** The database also exhibits a very high load average, indicating an over-capacity load.
- Bandwidth Utilization:** **There were no apparent network throughput bottlenecks in the system.** The bandwidth utilization averaged 6.8 Mbps at full load. (See *Detailed Findings, Observation 3*).

- **Transactions A** – <Brief business description of transaction >
- **Transaction B** – <Brief business description of transaction >
- **Transaction C** – <Brief business description of transaction >

Performance Targets

The performance criteria specified by the customer, which we tested for, are as follows:

- Target load: The target load the system should be able to handle is 300 users, in roughly the mix of transactions designated in Table 3.
- Page response times: The maximum time for a page response once the user invokes it is 4 seconds.
- Transaction completion times: Transaction B, which is the most complex transaction, must be able to be completed in 4 minutes or less, at the target load

Test Environment: The test environment was comprised on the following information provided by the customer:

Table 2 – System Configuration	
Component	Configuration
Web & Application Server	Intel 2x1Ghz, 1GB memory, RedHat Linux, 2x36 GB disk Apache and JBoss.
Database Server	One Compaq DL380 4x1Ghz hyper-threaded, 4 Gigabytes of memory; RedHat Linux, 3x36 GB disk, Oracle 9i
Network	One Alteon AceDirector3 Load Balancer, one Nokia IP330 Checkpoint Firewall, and a 100 Mbit internal network.

Recommendations

Based on our findings and conclusions, our recommendations are as follows:

Table 4 - Recommendations
<p>Recommendation # 1: <i>Transaction Tuning</i></p> <p>1. As the slowest performing page of Transaction A is page 4, we recommend tuning the code in that transaction, particularly any database queries it makes. Queries are often the slowest part of any application code, and can usually be optimized further.</p> <p>Benefits: In our experience, performance bottlenecks are attributable to the application code 80% of the time. Improving the performance of these transactions will increase the overall scalability of your system, enabling you to defer the expense of hardware upgrades.</p>
<p>Recommendation # 2: <i>Evaluate increasing the capacity of your database server</i></p> <p>Once you have implemented recommendation # 1, we recommend that you re-test to quantify the improvement, and if the database server still saturates, consider increasing its cpu configuration.</p> <p>Benefits: While database servers typically run at high cpu utilizations, you increasing the speed of the cpus will provide headroom for your expected growth in donor activity, and will ensure a positive user experience for potential donors.</p>

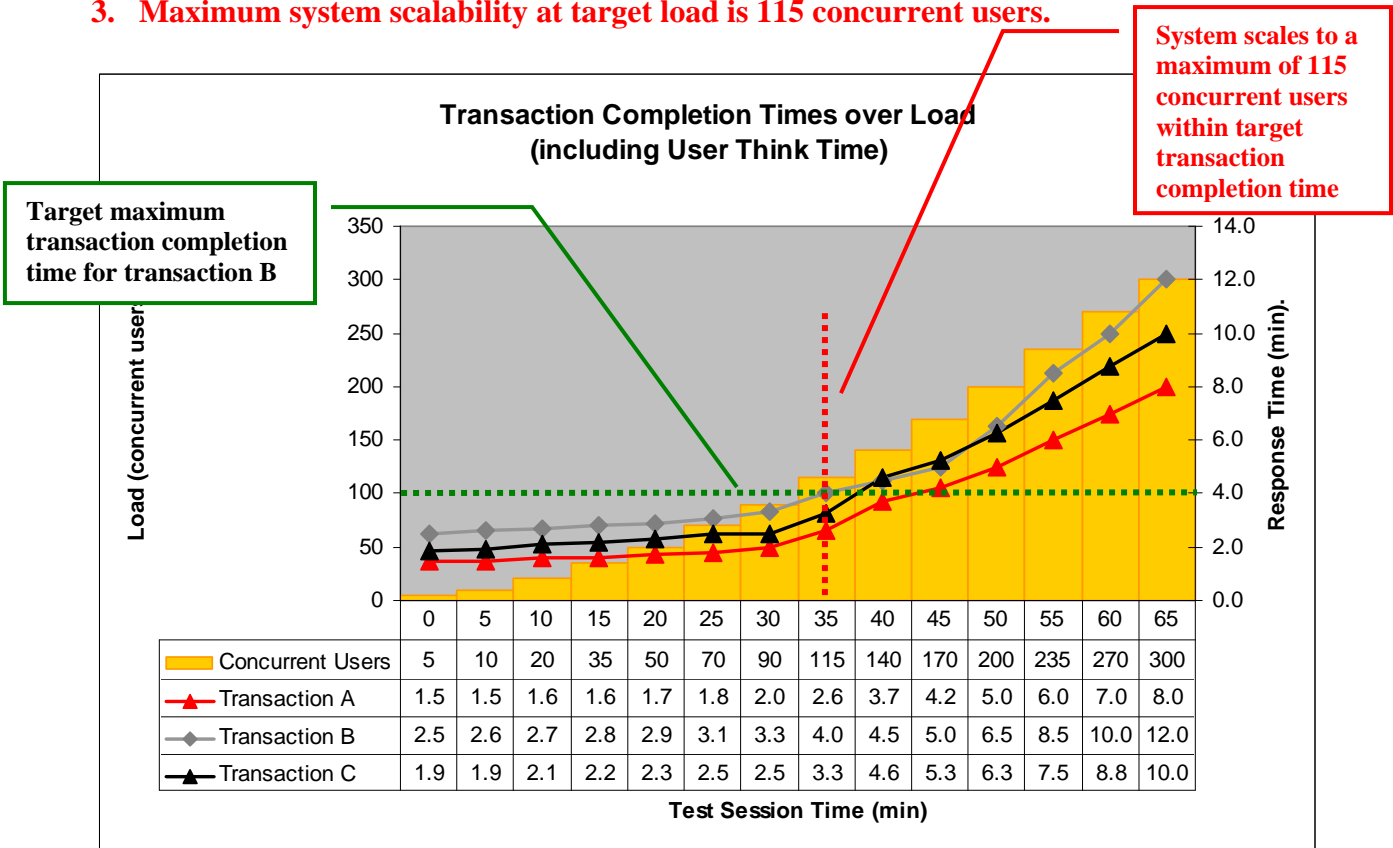
Detailed Findings

Observation # 1: System Scalability

Objective: Quantify the average transaction performance over load times for all transactions and identify the maximum number of concurrent users that can be supported within the target response time.

Observations:

1. The system shows a linear response up to about 25 concurrent users.
2. Gradual degradation occurs from 25 to 100 users for the two most resource-intensive transactions
3. **Maximum system scalability at target load is 115 concurrent users.**



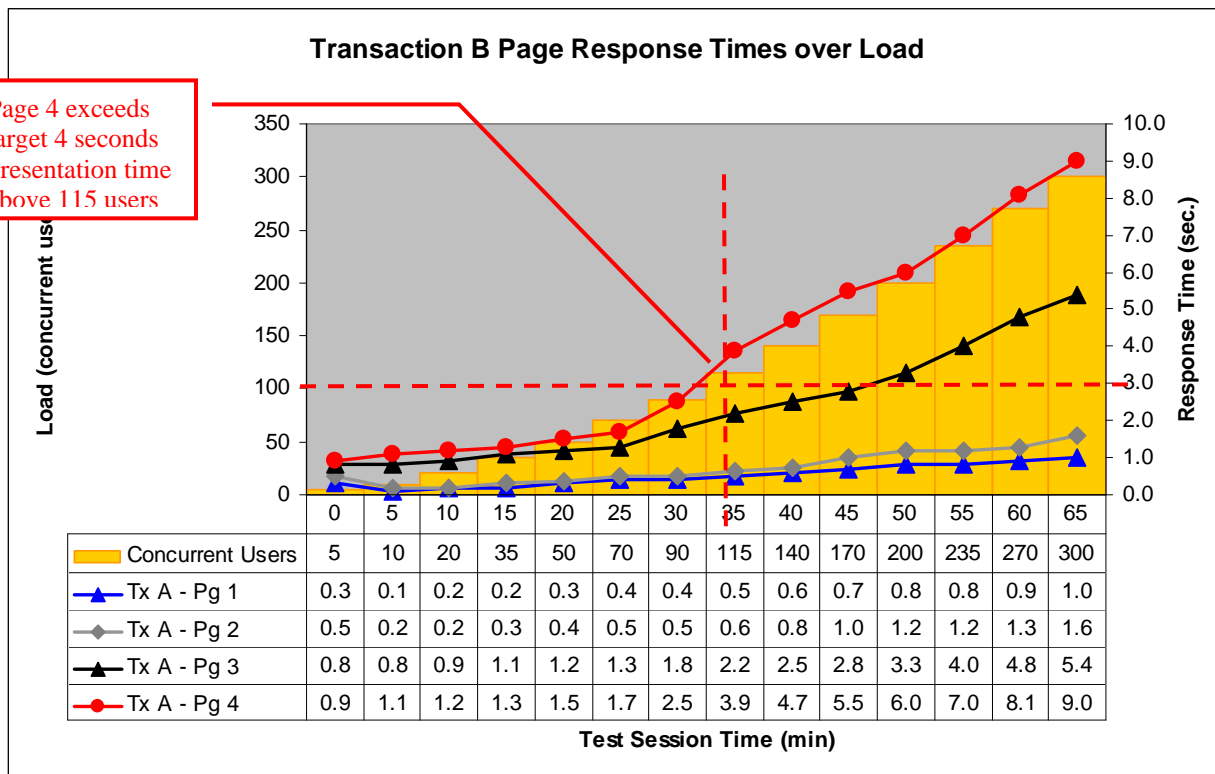
Observations # 2: Page Presentation Response

Objective: Quantify the average page presentation times for all transactions.

(Note: Shown are results for Transaction A only. A complete report would show detail for all relevant transactions.)

Observations:

1. Page 4 of Transaction B exceeds the 4-second maximum response time above 115 concurrent users. This is the page to focus on for subsequent tuning.
2. Page 3 begins to show substantial degradation at 90 users, but remains below the target threshold until it passes 235 concurrent users.
3. Pages 1 and 2 scale almost linearly throughout the load range, and are much less significant contributors to Transaction A’s performance.

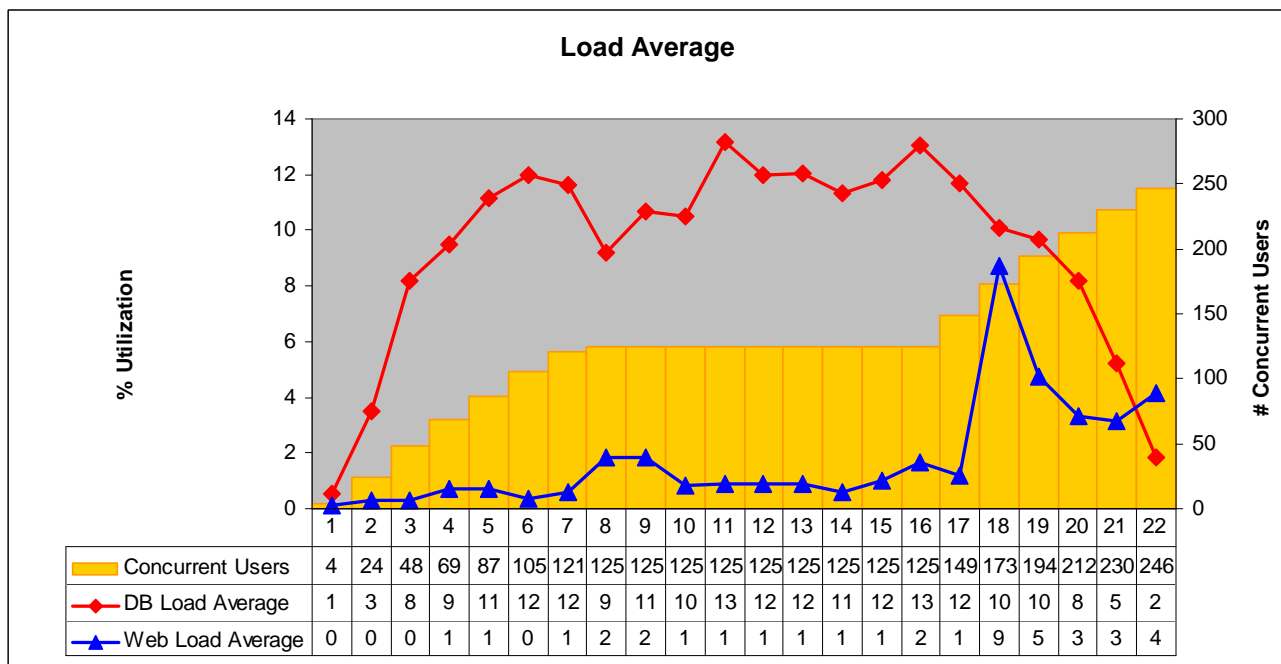
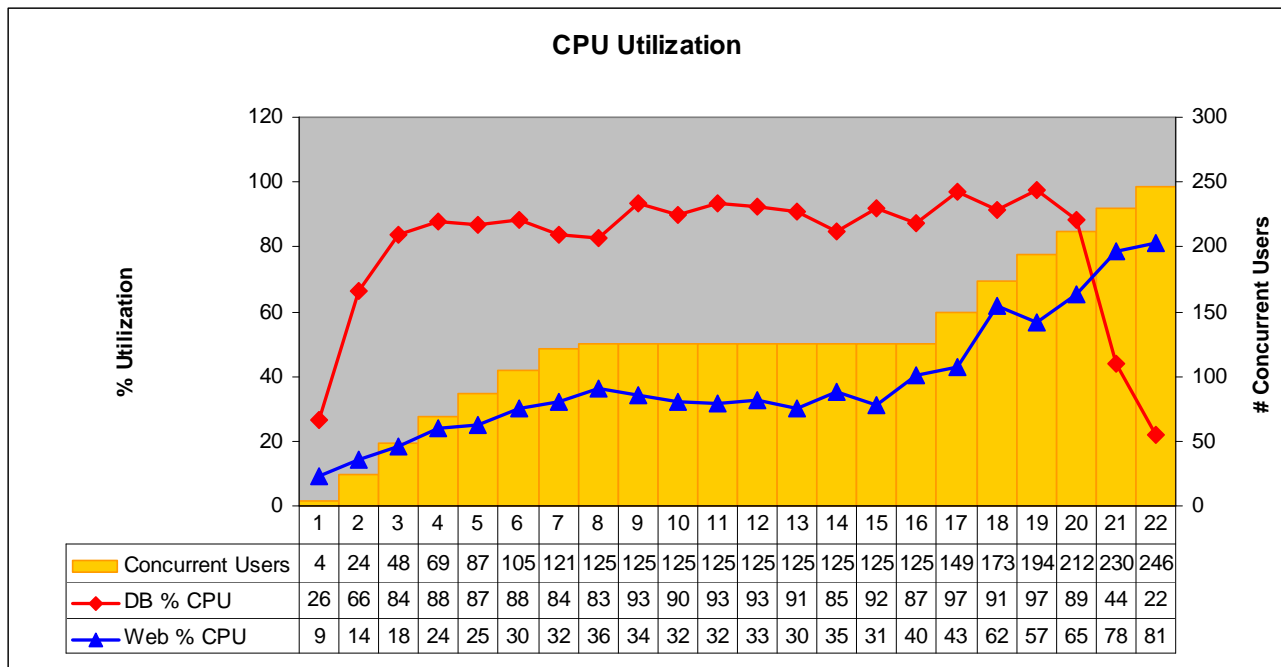


Observations # 3: System Resources

Objective: Identify key system resources that may be creating performance bottlenecks.

Observations:

1. Database cpu rises quickly to 88% within 6 minutes, stabilizes at 125 user load, then essentially saturates at 97% above 130 users.
2. The database load average rises sharply, staying in the 10-12 range throughout the load range.

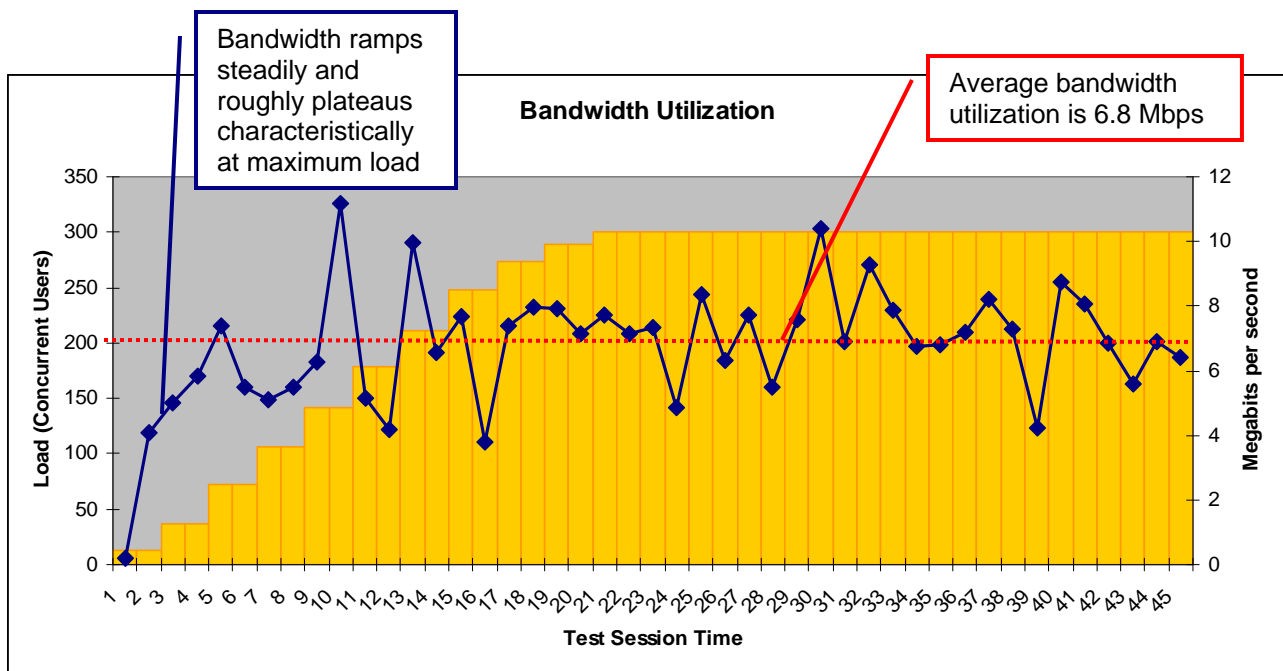


Observations # 4: Bandwidth Utilization

Objective: Quantify the total bytes sent and received and identify any bandwidth bottlenecks.

Observations:

- 1. As load was ramped from 0 to 300 concurrent users, and total bandwidth utilization (sent + received) increases steadily, and plateaus characteristically at full load. This indicates that there are no bandwidth bottlenecks imposed by the network or the system.
- 2. Bandwidth utilization averaged 6.8Megabits/second (Mbps). Data sampled at 1 minute intervals shows spikes of activity with spikes to 11 Mbps, typical for this sampling interval.
- 3. The negative spikes approaching full load and during full load can be accounted for by server response timeouts as evidenced by the rise of http error 500s.



Glossary of Terms

Term	Definition
Concurrent users	The number of user sessions actively executing a transaction at one time, but that at any given instant could be idle from the computing environment's point of view, due to a think-time or pacing pause, or waiting for the system to respond.
Iteration	A user's full cycle through the pages that define a transaction.
Pacing	The randomized pause interval specified for users between iterations. Pacing is specified to simulate the real load conditions both of a user pausing between one activity and the next and the randomized arrival of new users.
Ramping rate	The initial arrival rate of users as they start to execute their individual transactions. A ramping rate is specified to simulate real load conditions and to not immediately overwhelm the system by an unrealistic simultaneous arrival of the full target load.
Scalability range	The load range throughout which transaction response degrades at a relatively gradual rate. It is usually followed by a sharp increase in degradation rate, depicted by a response curve that turns visibly upward.
Transaction	A <i>business process</i> or <i>function</i> of an application, as defined by a specific navigation of pages and user actions.
Transaction completion time	The time it takes for a user to complete the page navigation that fulfills a transaction, including the embedded think-time.
Transaction volume target	The target number of each type of transaction that the system is expected to support at peak load.
User think-time	A pause between user actions in a process that simulates the time for the user to consider what to do on a page and do it.