A Model of a Web Server with Dynamic Content

Varsha Mainkar
Network Design and Performance Analysis Department
INFORMS Fall ‘99 meeting
Philadelphia, November 8, 1999
Typical Web-Based Service

Web Server Front End

Internet

LAN/WAN

Back End Systems
(Databases, Email Servers, Telephony servers, Legacy Systems)

Client PCs with Web Browsers

Web Server

(V.Mainkar, 11/8/99)
Typical Web-Based Service

1. Client sends HTTP* request

2. Thread is activated to serve the request

3. Retrieves dynamic info

4. Formats info into HTML* and returns result

*HTTP = HyperText Transfer Protocol, the protocol used for Web transactions. HTML = HyperText Markup Language, the formatting language for Web pages
Performance Measures of a Web-Based Service

Response time and blocking seen by clients

Internet

Web Server Front End

Web Server Capacity

Back End Systems
( Databases, Email Servers, Legacy Systems )

Modeled as a “black box”

Client PCs with Web Browsers

Web Server

LAN / WAN

V. Mainkar, 11/8/99
Server resources that are potential bottlenecks

- Number of TCP connection requests that are not yet accepted
- Number of active threads (servlets) in the Web Server
- Number of requests that can be waiting for a servlet thread

Network I/O subsystem may also be a potential bottleneck. However, in this talk we focus on “CPU-bound” applications.
Web Transaction Flow & Queueing Model

TCP Connection Request (SYN) → Web Server

HTTP Request (SYN-Ack) → Browser

Ack, HTTP Request → Servlet

Servlet = Thread spawned by a Netscape-type Web server, to handle dynamic processing
RTT = Round Trip Time

Queueing Model

TCP Connection
Queue: Multiple Server, no waiting room. M/M/c/0
Service time = Internet RTT

HTTP Queue:
Multiple Server, waiting room. M/M/c/K
Service time = Total time that a servlet is active
Queueing Model: CPU

Flow of typical servlet that generates dynamic content:

- Request for CPU: t1 secs
- Wait for I/O with back end system: w1 secs
- Request for CPU: t2
- Wait for I/O with back end system: w2
- Request for CPU: t3

CPU modeled as a processor sharing queue
Arrival rate of requests to this queue = Web transaction throughput rate
X number of CPU request segments in the servlet
Response time of a request segment of time t is \( t/(1 - a) \) where
\( a \) is ...
Hierarchical Queueing Model

Then, holding time of servlet is =

\[ w_1 + w_2 + \ldots + R_{\text{cpu}}(t_1) + R_{\text{cpu}}(t_2) + R_{\text{cpu}}(t_3) + \ldots \]

where \( R_{\text{cpu}}(t) \) is the response time of a request in the CPU queue.

Finally, model variables are interdependent, so iterate until convergence is achieved.

Implemented in Mathematica.
Performance Measures

• Web Transaction Response time:
  – TCP connection set up time + HTTP queue waiting time
    + servlet holding time + 0.5 x Internet RTT
  – TCP connection set up time = 1.5 Internet RTT

• Blocking:
  – Blocking at TCP queue ($B_{tcp}$) and at HTTP queue ($B_{http}$)
    • $B_{tcp} + (1 - B_{tcp}) B_{http}$

• Web Server Capacity: the transaction arrival rate
  at which a certain response time and blocking requirement is met
Model Validation

• Validation of this model was done against measurements on a simple test environment

• Test Environment:
  – Hardware: PC with 200 Mhz Pentium, 96 MB memory
  – OS: Windows NT 4.0 workstation
  – Web Server: Netscape Enterprise 3.6

• Web transaction:
  – A simple “test” servlet that uses the CPU for some time, then waits (sleeps), then uses CPU again, then waits…
  – Specifically: \( t1 = t2 = t3 = t4 = 2.1 \) seconds.
  – And \( w1 = 1 \) sec, \( w2 = 2 \) secs, \( w3 = 3 \) secs
Model Validation

- Measurements were done using traffic generated by Silk Performer, using 1-11 users
- The following was measured
  - Average response time
  - Blocking percentage
Test vs Model: Scenario 1

- Tests on a LAN, Web Server thread limit = 512
- Internet RTT ~ 0
Test vs Model: Scenario 2

Tests on a dial up line (Internet RTT ~ 140 ms), Web Server thread limit = 512

Response Time vs Arrival Rate - dialup test, 512 thread limit

Arrivals per hour

Response time (seconds)
Test vs Model : Scenario 3

- Tests on a dial up line, Web Server thread limit = 3
- HTTP waiting room size: unknown
- Model can be used to estimate that size

**Response Time**

![Graph showing response time vs hit rate for different models](image)

- **Test**
- **Model - no HTTP queue**
- **Model - HTTP queue size = 6**
- **Model - HTTP queue size = 100**

---

V.Mainkar, 11/8/99  14

AT&T PROPRIETARY
USE PURSUANT TO COMPANY INSTRUCTIONS
Test vs Model: Scenario 3

- Tests on a dial up line, Web Server thread limit = 3
- HTTP waiting room size: unknown
- Model can be used to estimate that size

![Blocking vs Hit rate - dial up test, thread limit = 3](image)

- Model - no HTTP queue
- Model - HTTP queue size = 6
- Test
- Model - HTTP queue size = 100

Arrival rate per hour vs Response time (seconds)
Conclusions

• Simple testing shows promising results -
  – Although model was simple, the model results were acceptably close to test results
  – There is a lot of room for improvement, which should result in closer estimation of measurements

• Modeling can help in quick prediction of performance even when parameters of Web Server or OS software are not known