HATS 7.1 Performance and Capacity Planning

In this article we highlight the processes to improve the performance and capacity of our HATS applications. We also discuss techniques to build efficient HATS applications and how to isolate performance issues- Where problems occur? Best performance practices and tools and where to look for help.

Performance Issues:

There are a few areas where issues mostly occur. A few guidelines are provided below to tackle them:

**Browser settings**

There two important areas that can be tuned to improve performance namely, cache and HTTPS.

Enable browser caching whenever possible to cache these files

- KBS.js (28 KB)
- lgwfunctions.js (112 KB)
- Images
- -.css (average 2.2 KB)

Using HTTPS will cause extra work at the HTTP server and thus may affect capacity. H

**Network Settings**

- Link speeds, Routing and MTU are the areas that need to be watched out here.
- Nagel Algorithm is a client option in HATS and it should be disabled.
- Most IP Stacks have a delay as a left over from older slower networks. DelayAcks should be disabled for high levels of performance:

zOS Example: PORT 8082 TCP NO_DELAYACKS

**Operating systems & Telnet server**

Operating System and TCP/IP stack
There are many tuning parameters that affect performance at the operating system and TCP/IP stack level.

- Refer to the Performance and Capacity Planning Guide for your platform.

- Enable contention resolution on Telnet server

**Webserver**

Tune HATS Projects

- Read and understand Appendix A Screen-settling Reference from the HATS User’s and Administrators Guide.

- Strategies

Timing:

Used when the host type is TN3270, and when the host type is TN3270E and contention resolution is not used.

- Fast3270

Used when the host type is TN3270E and the contention resolution feature is used on the connection.

- Fast5250

Default for 5250 connections, and is usually valid

**HATS**

- HATS HTTP compression

- Introduced in HATS v6.0.2

- Transformation content only (average 80%)

- Not available for portlets

- Does not compress *.js or *.css files

- Significantly reduces size of HTTP data stream

- Requires browser that supports GZIP compression

- Web server compression

- Compresses style sheets, JavaScript files, gifs, etc. (average 75%)
HATS Transformation Settings

- Turn Keyboard On/Off

Place button on the Application Keypad, by default option is on Pressing Ctrl-k performs the same result. Turning off the button on the Keypad results in approximately 2% better performance at high CPU utilizations.

- Function key recognition

- Visual enhancement that does not disable the keyboard if users will use the keyboard, disable the function keys. Saves approximately 3% performance at high CPU utilization.

- OIA rendering

For Web usage not recommended, inconsistent with a Web application.

Only enabling Input inhibited indicator, system locked and message waiting indicator (5250 only) - HATS performs approximately 4% better at high CPU utilizations. Disable OIA completely – HATS performs approximately 6% better at high CPU utilizations.

- Selection list component

HATS performs approximately 7% better at high CPU utilizations with this function disabled. Generally a good option to use, but understand the effect at high CPU utilization.
Performance Tools:

zSeries Technologies

- Hipersockets™
- When using CS/Linux on zSerie
- Fast memory-to-memory transfers

zSeries Application Assist Process

- Java runs on the zAAP
- Frees cycles from CPs
- Can increase total throughput

Encryption Acceleration

- Does SSL on external engine
- Frees cycles from CPs
- Can increase total throughput.

HATS Tracing

Enable runtime tracing

• trace.RUNTIME=7
• trace.HOD.PSEVENT=1
• trace.HOD.OIAEVENT=1
• maxTraceFileSize=102400

- Ethereal trace formatter

For analyzing macro efficiency

Requires the following:

- Active Perl:
  http://www.activestate.com/store/languages/register.plex?id=ActivePerl

- Ethereal: http://www.ethereal.com/distribution/win32/)
Merges HATS tracing information with the Ethereal trace

**Web Server Monitor**
- Use when addressing response time issues
- Available for Apache and IBM HTTP servers
- Technote available on how to set up and use
  • Search WebSphere Technotes
  • “Monitoring IBM HTTP Server connections”

**Tivoli Performance Viewer**
Shipped with WAS Admin
• Must enable Performance Monitoring Service
• Will show HATS response times by servlet
Change processor speeds - xSeries

The following methodology is an estimate. Consult your hardware specialist for a more accurate estimate. Increase in processor speed is not linear

- i.e. Doubling processor speed of a server doesn’t double the capacity

- Theoretical capacity increase (z) = y/x - 1, where

  • x = speed of measured processor
  • y = speed of desired processor (be sure to use same frequency for both, e.g. GHz)

- Realistic capacity increase (w) = z/2

- Capacity of server c2 = (1+w) * c1

- Example:

  • c1 = 2-way 2.0 GHz (x=2)
  • c2 = 2-way 3.0 GHz (y=3)
  • z = 3/2 – 1 = 0.5; w = 0.5/2 = 0.25
  • c2 = (1+0.25)*(2) = (1+0.25) * 2000 users = 2500 users

Where to Go for Help

- Techline does HATS sizings:
  http://w3-03.ibm.com/support/americas/techline/sizewise.html

- If you are having performance problems:

  Check tuning.
  
  Check connections and messages with HATS Administration.
  
  If issue is not resolved, open a PMR to further diagnose.

Reference: https://extranet.lotus.com/rationalhats721