ServiceCenter 6.2
Horizontal Scaling in Clustered Configurations

Updated Technical Bulletin

February 2008

February 2008

Table of Contents

Overview

This document is an update to the information provided in September 2007 regarding horizontal scaling issues in ServiceCenter 6.2. This will provide an update on the technical aspects of the issue, the changes that have been made, the validation process and our delivery plan.

In September 2007, HP communicated details about a design issue that affects ServiceCenter 6.2, specifically in applications running ServiceCenter Run Time Environment (RTE) version 6.2. This design issue prevented the application from scaling to an acceptable level when in a horizontally scaled (clustered) environment. Over the past few months HP has completed a comprehensive development and test effort to address the issues.

HP is now recommending that customers who wish to deploy this technology should do so. The known issues have been addressed and a comprehensive test program has been completed.

Servlet Implementation

Vertical scaling with servlets on a single host has been tested successfully in many different situations and configurations. The servlet implementation has been shown to be reliable. The known issue regarding horizontal scaling should not be confused with the servlet technology in general. This issue is specific to the horizontal scaling feature.

Technical Details

The fundamental issue is isolated to the way in which resource management is handled across the cluster. Over a period of time, the system becomes unable to keep up with the demand for shared resources. This manifests itself in eventual response time degradation, and can also lead to session failures. This issue manifests itself mostly in load testing activities when there is a high level of concurrence (that is, many users doing the same activity at high volumes).

Known Issues with Horizontally Scaled System Implementations

The following is a description of the processes that can cause over-demand of shared resources across the cluster:

1. Multiple threads within the same process could all request the same resource under situations of high concurrency. This results in very high level of multicast messages for the same resource.
2. When multiple threads attempt a lock that is not available, all threads continue to retry the same request simultaneously.

The combination of these two processes led to an abundance of “chatter” within the cluster. The system could then become overwhelmed by trying to maintain this volume resulting in poor performance or failure.

This issue was only seen in horizontally scaled implementations. It is not applicable to vertically scaled systems because in vertically scaled systems resource management is local. Local system resource management eliminates the need for intersystem communication, which was the primary contributing factor to this issue.

Changes Made to Address Known Issues

The following changes have been implemented in order to address the known issues with horizontally scaled system implementations:

- To address issue 1 above, a queue was implemented so that only one thread will request a resource at a time.
- To address issue 2 above, a prioritization scheme was implemented. This scheme will randomly assign a priority to each process. The highest priority (lowest number) process will be allowed to resolve its resource requests first, and then others will follow in priority order sequence.
- Additional diagnostic capabilities were added to help quickly identify future problems that may arise. These options should be used with the guidance of HP Customer Support.
    - The debugrs:1 option will track the life span of a lock request and write a notification message to the sc.log file if a lock takes longer than 5 seconds to be satisfied.
    - The reportlbstatus option can be run at a regular interval to provide detailed information about the state of the cluster and a list of threads waiting for locks.

Version and Patch Availability

The changes described above are now available for ServiceCenter 6.2, beginning with patch release 6.2.4.2. Use this patch level or above to obtain the described changes.

Service Manager customers should use version 7.01 to obtain all the most current updates to the product as of this writing.

Contact customer support if you have any additional questions.

Test Program History

HP completed a comprehensive test program to validate the product changes detailed above. This was completed over the course of the last few months and was a high priority within the R&D team.

The testing consisted of functional testing, load testing, and customer system testing. The functional testing focused on reproducing the problem in a controlled environment with a previous release (SC 6.2.1). The tests were monitored to a point of failure, and also profiled with development tools to identify the pattern of failure. Those tests were then repeated with the proposed changes to verify that the failure condition no longer occurred.

Load testing was conducted to simulate high concurrency on configurations which represent the majority of the desired customer implementations. The tests were run in the lab using HP Load Runner to drive the necessary load. Test results are summarized below.

HP also worked closely with selected customers to test the changes in their environment. Due to the aggressive timeline of the test program, HP did not have enough time to complete lengthy test programs with all partner customers. The testing completed to date has not raised any problems or issues.

Test Program Results

Sample Test 1

HP-UX, 600 users/3 machines

Machine 1: 2 CPU (Itanium IA64 1.6GHz) x 4 GB Ram, 3 servlets and the load balancer.

Machine 2: 2 CPU (Itanium, IA64 1.6GHz) x 4 GB Ram, 3 servlets + IR async process. (Asynchronous processing of IR index updates is used instead of synchronous processing to improve performance).

Machine 3: 2 CPU (Itanium, IA64 1.6GHz) x 4 GB Ram, 3 servlets

3,000 incident tickets opened in 1 hour (5 tickets per user).

Response times: - Login, avg=2.046s; Open Ticket, avg=2.723s

Sample Test 2

Windows 2003 Server, 2200 users, 3 machines.

Machine 1: 8 CPU (dual core 2.83GHz) x18 GB Ram, 10 servlets and the load balancer

Machine 2: 8 CPU (dual core 2.83GHz) x18 GB Ram, 10 servlets + IR async process

Machine 3: 2 CPU (dual core 2.83GHz) x 8 GB Ram, 3 servlets

11,000 tickets opened in 1 hour (5 tickets per user).

Response times: - Login, avg=2.256s; Open Ticket, avg=1.442s

Sample Test 3

(Vertical scaling test as a baseline)

Solaris 10, 500 users, 1 machine

Machine 1: V440 4 CPU (Ultrasparc IIIi, 1.593GHz) x 8 GB Ram

2,500 tickets opened in 1 hour (5 tickets per user).

Response times: - Login, avg < 3 seconds; Open Ticket avg < 3 seconds

February 2008