

IBM Tivoli Security Compliance Manager Version 5.1

Performance Report

The information in this report is IBM/Tivoli Confidential. It can be shared with IBM/Tivoli customers that have signed a Non-Disclosure Agreement or Confidential Disclosure Agreement. Customers without such agreements should not have access to this report. Be careful how the information in this report is used. Taken out of context, performance information can be misleading.

This report contains internal performance measurements of IBM Tivoli Security Compliance Manager Version 5.1. These measurements should be used for planning purposes, but note the following. They are derived in a test lab, based on specific configurations that are described herein. They are only a guideline for estimating performance. Performance results in a production environment will vary based upon the specific parameters and uniqueness of that environment.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A.

(C) Copyright International Business Machines Corporation 2004. All rights reserved Note to **U.S. Government users** -- Documentation related to restricted rights -- Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp.

Document Version Number: 1.2
Document Creation Date: 5.30.04
Document Update Date: 6.24.04
Authors: Levoy Wheeler
Status: **IBM Confidential**

Table of Contents

Table of Contents	iv
Document History	Error! Bookmark not defined.
Introduction	5
Performance Test Procedures	6
Tools	6
Measurements.....	7
Performance Test Hardware	9
Performance Test Results and Observations.....	10
Message Throughput Load.....	10
Message Throughput Calculation	15
Client Update Load	15
Proxy Impact.....	18
Client Long Term Impact	19
CLI Response Times	20
GUI Response Times.....	20
Hardware Specifications and Recommendations	22
Hardware Specifications and Recommendations	22
Notices	24

Introduction

This document contains the performance test procedures, results, observations, and recommendations for IBM Tivoli Security Compliance Manager components. This document is to be used for deployment planning. These recommendations are based on performance benchmark testing.

Note that performance measurements are derived in a test lab, based on specific system configurations. All performance numbers are to be used as a guideline for planning purposes only. Actual production numbers will vary based on uniqueness of the deployed environment.

Performance Test Procedures

Tools

System Information scripts

The system information scripts are simple shell scripts which execute a number of system commands to gather current system data. These scripts repeatedly execute the system commands and save the information to a text file. This data can later be analyzed by the user or correlated by the script output parser program. The following is the list of programs utilized by the scripts per OS:

AIX – netstat, iostat, vmstat, ps, sar, du, lps

Windows – typeperf (retrieves information from various system counters)

Solaris – netstat, iostat, vmstat, ps, sar, du

Linux - vmstat, ifconfig, iostat, ps, du

Script Output Parser

The script output parser is a Java program that accepts input from the file produced by the system information scripts and parses out four main measurements: processor usage, network usage, disk usage, and memory usage. The program also accepts normalized maximum values in order to calculate the measurements as a percentage of the supplied values.

Microsoft Excel

The Microsoft Excel spreadsheet program is used to create graphical representations of the output generated by the script output parser program.

Client Multiplier

The client multiplier program is a loading tool created from the actual SCM client. This program allows the user to emulate large numbers of clients on one machine.

Policies

SCM policies contain collectors and their corresponding parameters and schedules. These policies are utilized to create messages that will be duplicated by the client multiplier to create the necessary load. The collector message sizes ranged from a low of 600 bytes to 220 KB; the average size of all the collector messages was 12 KB.

Measurements

Message Throughput Load Measurement

The SCM client is designed to send messages, generated from the running of collectors, to the server. Although the client sends and requests other data, such as updates, collector messages will normally make up the bulk of the traffic between the server and the clients. This measurement is designed to determine the optimal message throughput for the server on the measured hardware.

The primary tool used to execute this measurement is the client multiplier. The client multiplier is set up to generate the equivalent traffic of a pre-determined number of clients at varying frequency. For all of these measurements the number of emulated clients was fixed at 10000; the frequency of the collector messages was varied from a low of 0.5 messages per second to a high of 16.7 messages per second.

During setup, the system information scripts were set to continually run throughout the execution of the tests. After the test interval was completed, the output of the system information scripts were saved and passed through the script output parser to produce data that could be easily graphed. Finally, Microsoft Excel was used to produce the graphs representing the system impact.

Client Update Load Measurement

The Client Update feature allows the SCM server to distribute updated client program files to all or some of the clients. These program files can become fairly large and impose a significant load on the server as well as the network in general. This measurement seeks to quantify the impact of this event.

The primary tool used to execute this measurement is the client multiplier which has a provision that allows it to emulate multiple client update requests and accept the response. Due to the nature of the client update architecture, the traffic is throttled by the number of clients as opposed to the server. Therefore, the client multiplier is set up to emulate a varying number of clients each of which requests the client update at its due time, creating a client update request frequency.

During setup, the system information scripts were set to continually run throughout the execution of the tests. After the test interval was completed, the output of the system information scripts were saved and passed through the script output parser to produce data that could be easily graphed. Finally, Microsoft Excel was used to produce the graphs representing the system impact.

Policy Distribution Load Measurement

SCM policies contain collectors and their corresponding parameters and schedules. When the user assigns policies to client groups, the collectors and schedules associated with the policies are distributed to the various clients. This results in an increase in

network traffic as the clients request the collector JAR files. This process can impose a significant load on the server as well as the network in general. This measurement seeks to quantify the impact of this event.

The primary tool used to execute this measurement is the client multiplier which has a provision that allows it to emulate multiple client collector download requests and accept the server responses. Due to the nature of the client update architecture, the traffic is throttled by the number of clients as opposed to the server. Therefore, the client multiplier is set up to emulate a varying number of clients each of which requests the collectors corresponding to the distributed policy.

During setup, the system information scripts were set to continually run throughout the execution of the tests. After the test interval was completed, the output of the system information scripts were saved and passed through the script output parser to produce data that could be easily graphed. Finally, Microsoft Excel was used to produce the graphs representing the system impact.

Proxy Impact Measurement

The SCM proxy relay is used to reach clients that reside behind a firewall. The proxy relay is a special collector that runs as a thread forwarding connection requests to the appropriate clients. This measurement seeks to gauge the impact of supporting 1000 clients through a single proxy relay.

Client Long Term Impact Measurement

In the long run, the purpose of a client is to execute collectors at specified intervals and return the information to the server for analysis. To this end, the client is designed to have a minimal impact on system resources. This measurement attempts to gauge the impact of a client in a permanent deployment.

CLI / GUI Response Time Measurement

The user interfaces to the SCM system are through the SCM CLI and SCM GUI provided in the Administration Utilities installation. These facilities contain programs which allow the user to fully administer the SCM system through a graphical interface or command line programs useful in scripting. These measurements gauge the response time of frequently used operations.

Performance Test Hardware

AIX

IBM pSeries 610
Model: 7028-6E1
Processor: 64-bit POWER3-II, 450Mhz
Memory: 2GB
Disk: 70GB
SPECInt: 313 base

Microsoft Windows

IBM NetVista
Model: 2292-42U
Processor: Intel Pentium 4, 2.2Ghz
Memory: 1GB
Disk: 74GB
SPECInt: 798 base

Sun Solaris

SunFire V120
Model: 380-0538-02
Processor: UltraSPARC Ili, 650Mhz
Memory: 2GB
Disk: 36GB
SPECInt: 230 base

Linux

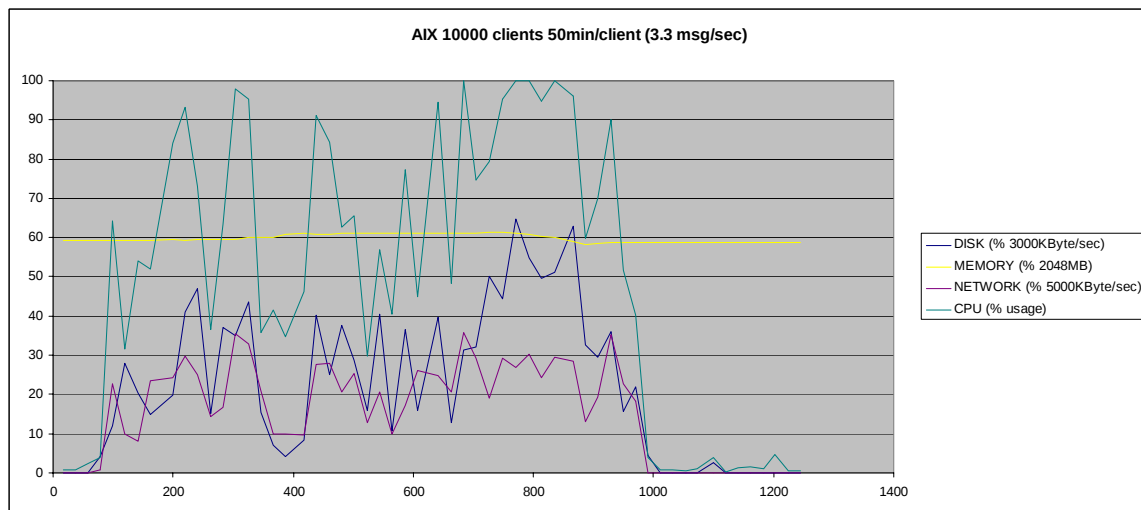
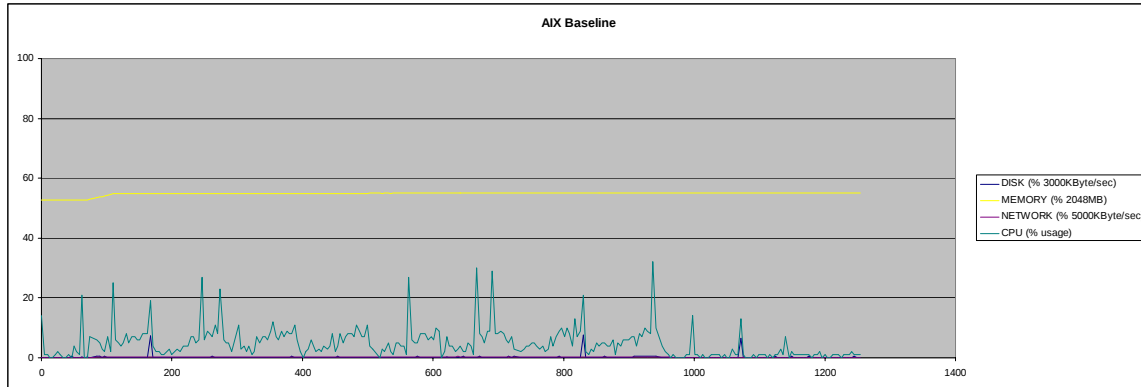
IBM NetVista
Model: 8305-4DU
Processor: Intel Pentium 4, 2.4Ghz
Memory: 2GB
Disk: 36GB
SPECInt: 940 base

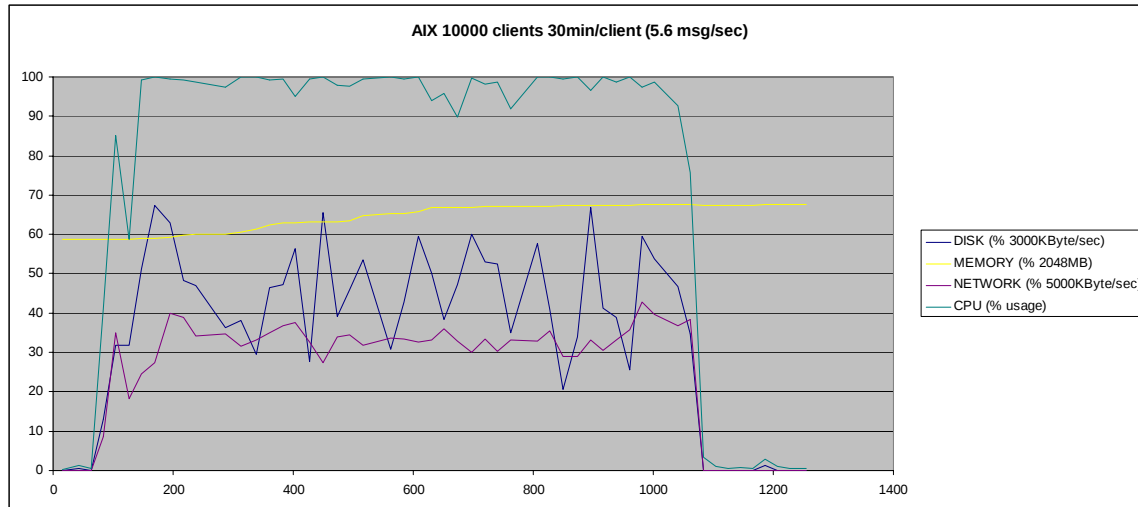
Performance Test Results and Observations

Message Throughput Load

AIX

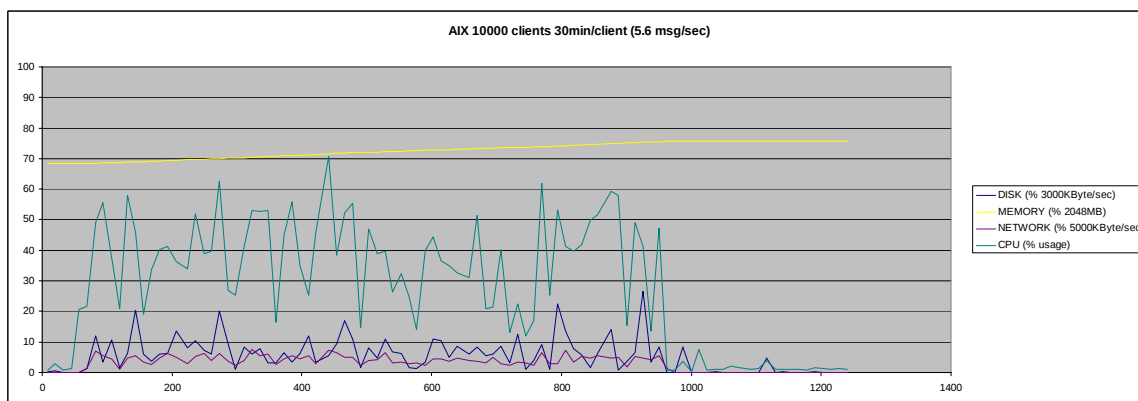
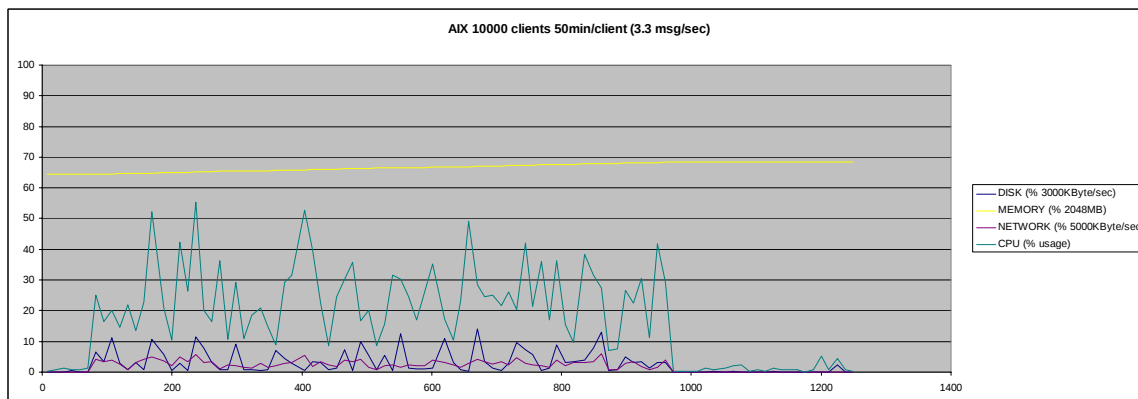
Local Database





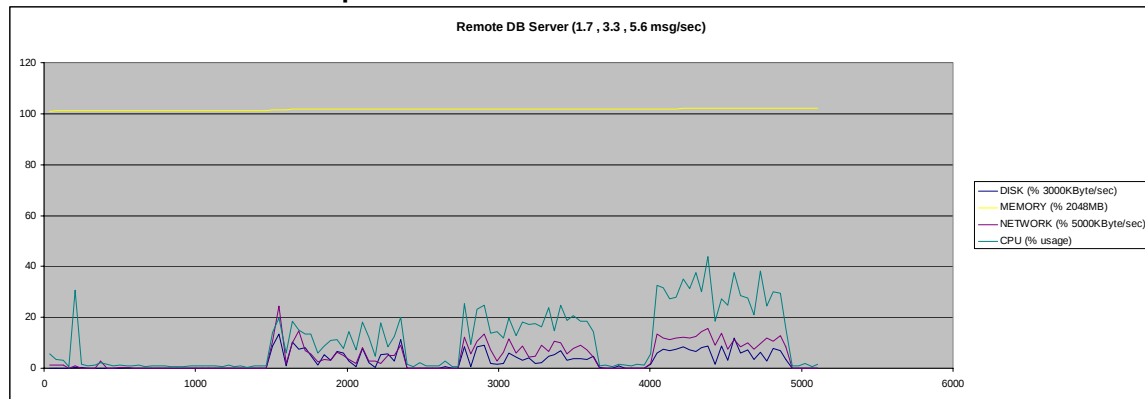
At 3.3 messages/sec the CPU usage averages around 70-75 percent, which is recommended. This allows some CPU resources to remain available for administrative tasks and message spikes. The CPU usage peaks out when the load approaches 5.6 messages/sec.

Remote Database



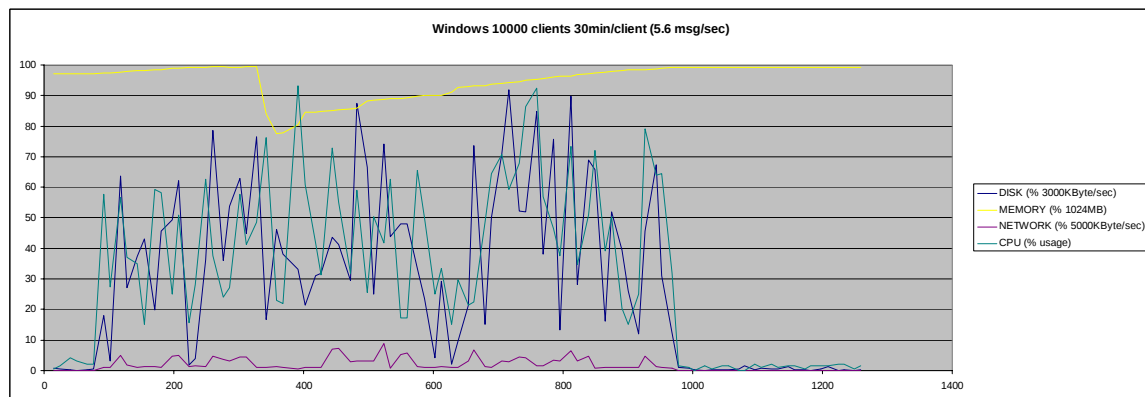
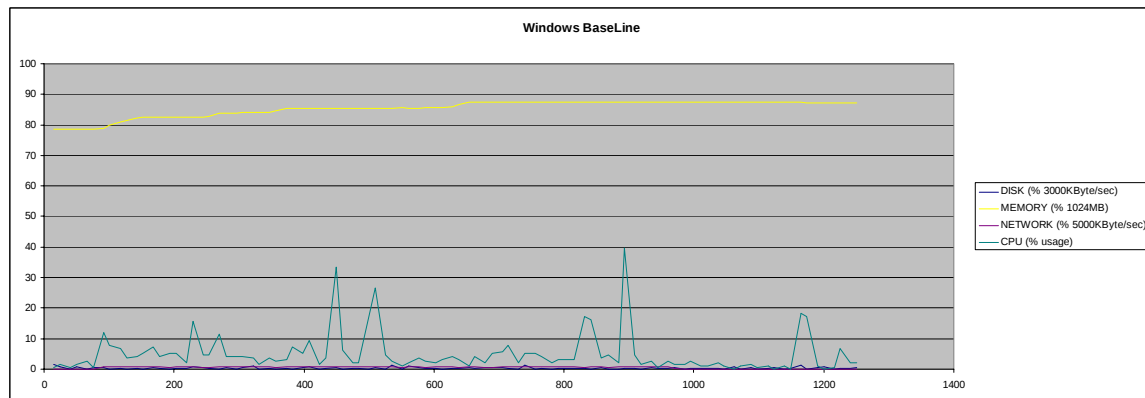
For the remote database setup, the CPU usage remains fairly low even at the 5.6 messages/sec load that peaks the local database setup. For the 3.3 message/sec load the CPU usage averages around 30 percent versus 70 percent for the local database.

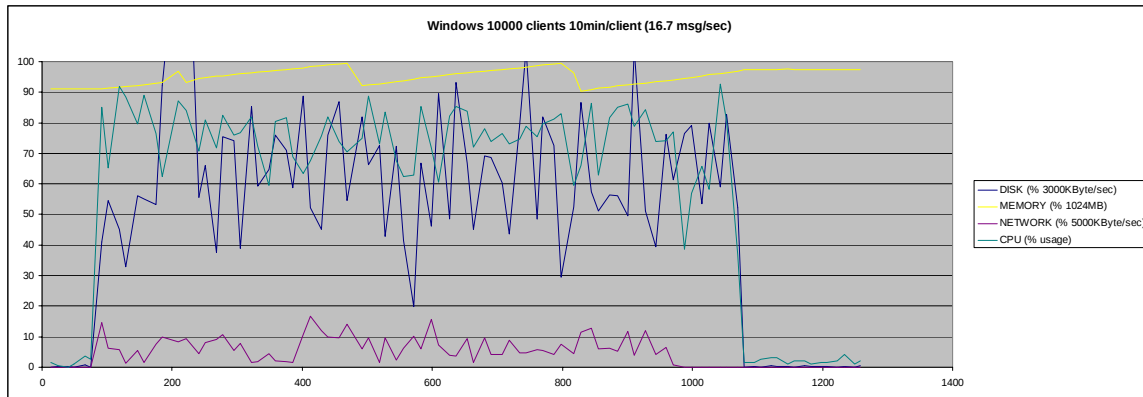
Remote Database Server Impact



The remote database server load remains acceptable during the three loading periods: 1.1 messages/sec, 3.3 messages/sec, and 5.6 messages/sec.

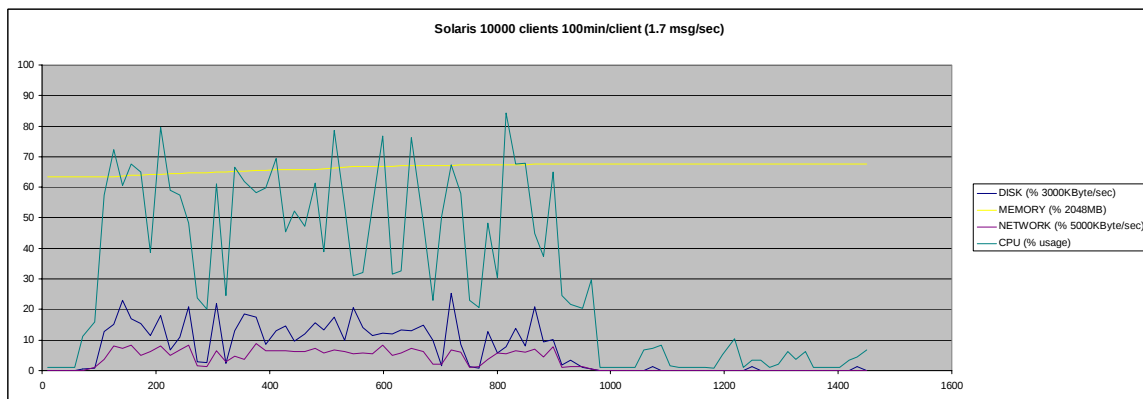
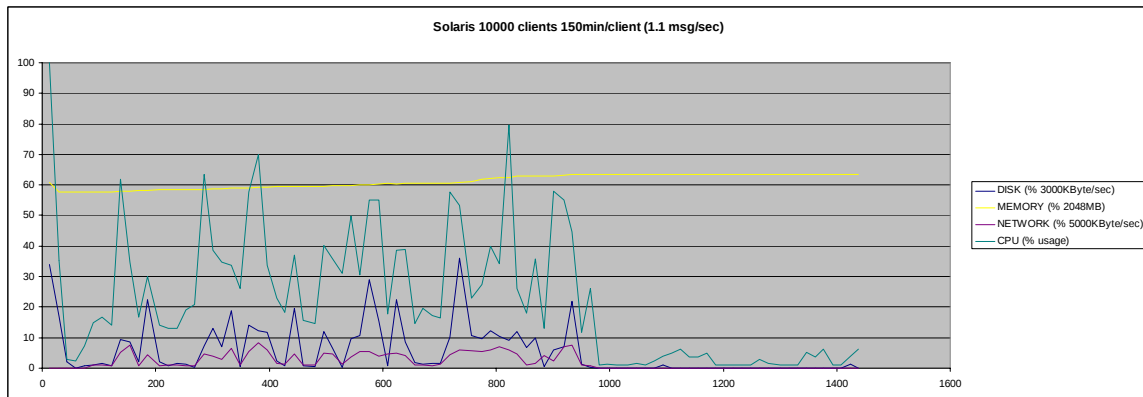
Microsoft Windows

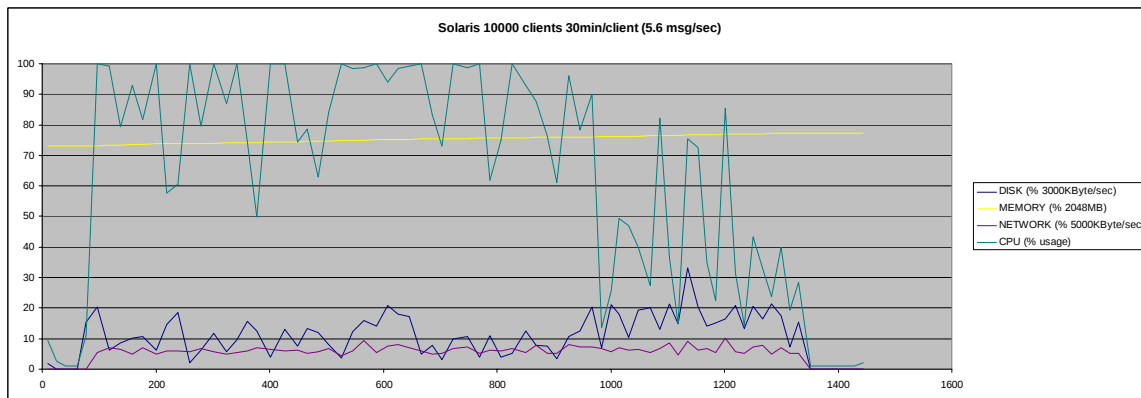
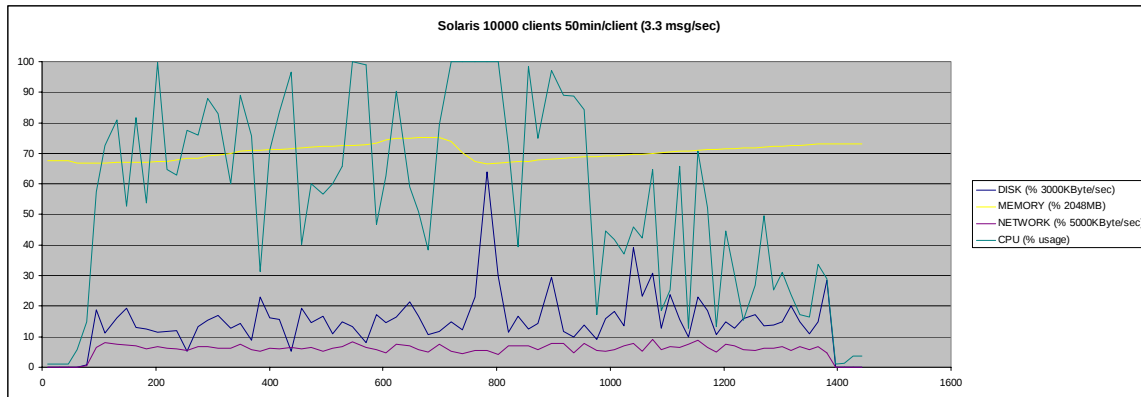




The Microsoft Windows server averages 75 percent CPU usage at 16.7 messages/sec.

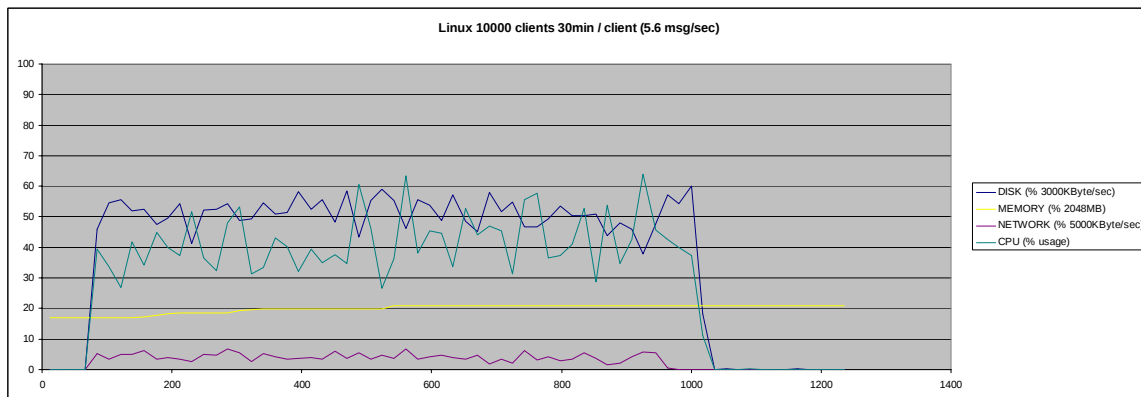
Sun Solaris

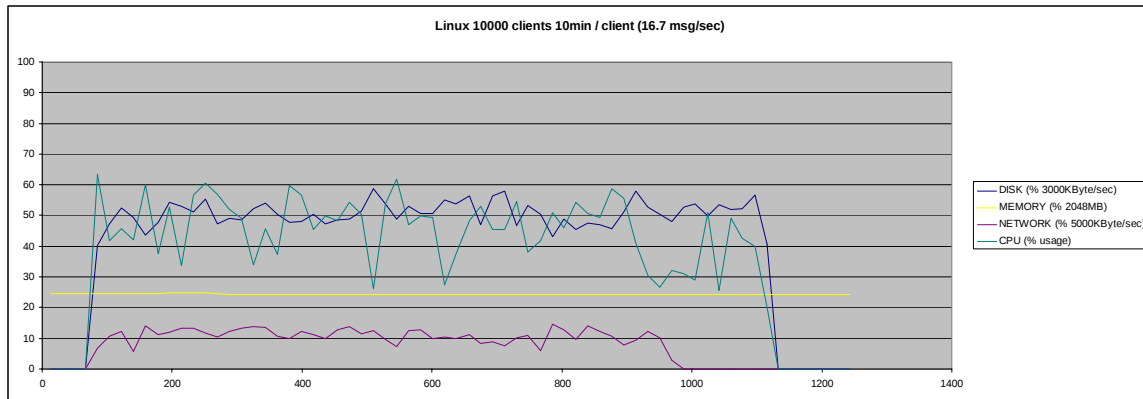




The Sun Solaris server peaks at 5.6 messages/sec. The Sun Solaris server establishes a 75 percent cpu usage at 3.3 messages/sec.

Linux





The Linux server establishes a mere 50 percent usage at 16.7 message/sec load.

Message Throughput Calculation

The message throughput, given the number of clients and collector schedules, can be calculated using the following formula:

$$N * (X/(604800 \text{ seconds/week}) + Y/(86400 \text{ seconds/day}) + Z/(3600 \text{ seconds/hour})) = M$$

Where:

N – the number of registered clients

X – the number of collectors on one client that are scheduled to run on a weekly basis

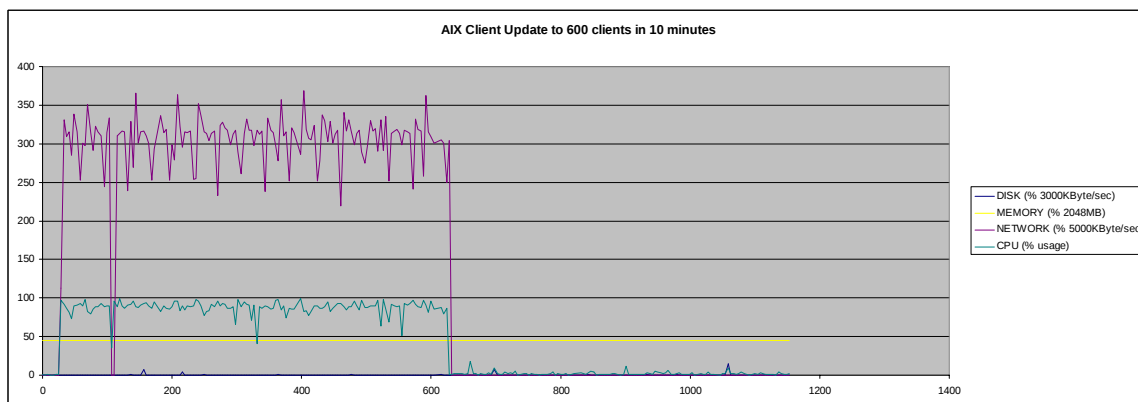
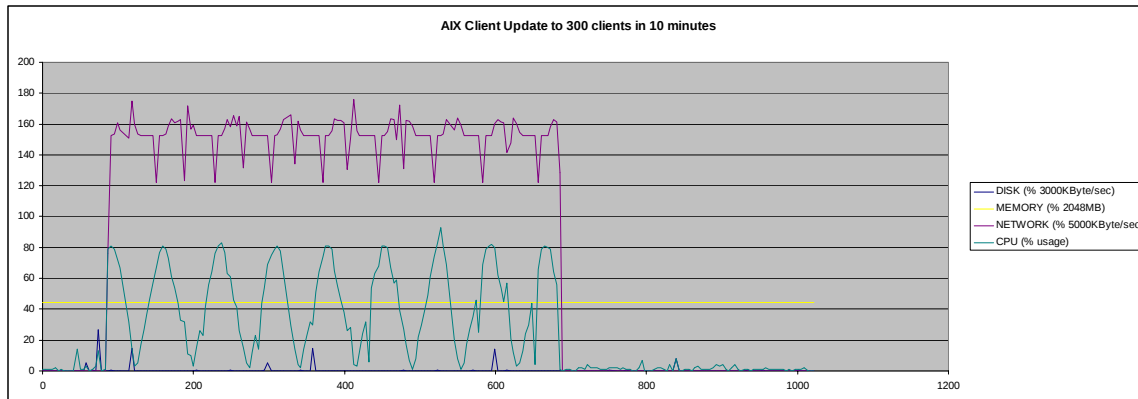
Y – the number of collectors on one client that are scheduled to run on a daily basis

Z – the number of collectors on one client that are scheduled to run on an hourly basis

Number of Clients (N)	Weekly Collectors/Client (X)	Daily Collectors/Client (Y)	Hourly Collectors/Client (Z)	Messages/Sec Throughput (M)
1000	0	24/86400	0	0.28
10000	24/604800	0	0	0.40
10000	0	24/86400	0	2.78
10000	0	20/86400	4/3600	13.43
10000	0	12/86400	12/3600	34.72
10000	10/604800	10/86400	4/3600	12.43

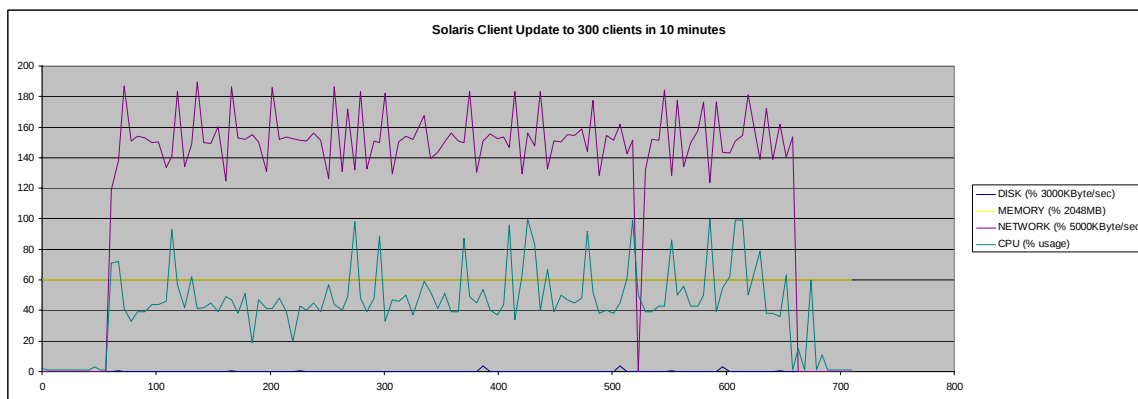
Client Update Load

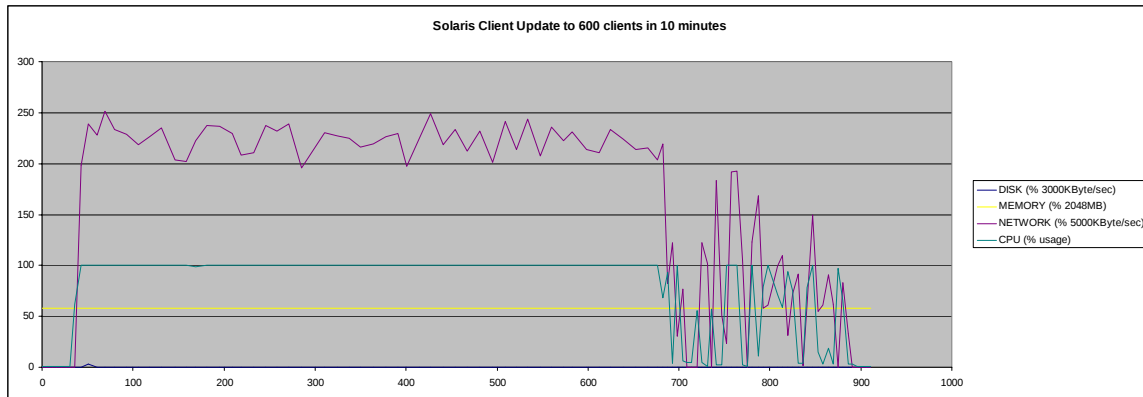
AIX



It is assumed that during client updates the server will still be accepting client messages at some rate, therefore, it is recommended that the additional load imposed by the client updates be kept below the 30 percent surplus CPU resources available at the recommended messages loads. The AIX system averages about 40 percent at 300 clients / 10 minutes. The default client update interval is 4 hours. At a frequency of 300 clients / 10 minutes the default setting would support 7200 clients (4 hrs * 60 min / 10 min * 300 clients). Again it is recommended that the average load be kept below the 30 percentile.

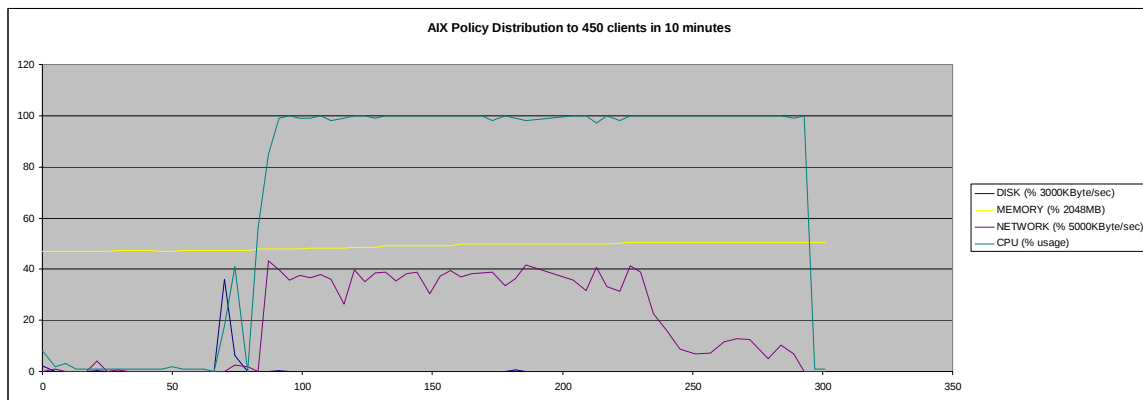
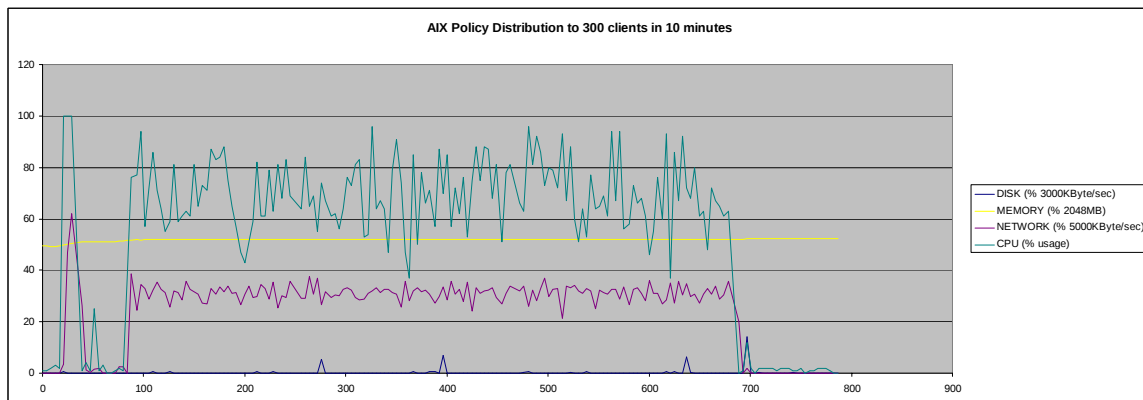
Solaris





The Solaris server averages around 50 percent CPU usage at 300 client updates / 10 minutes.

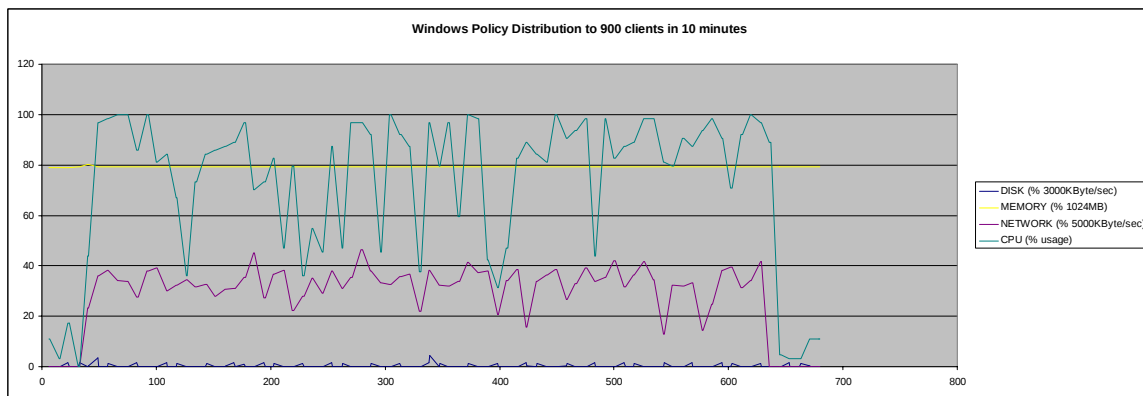
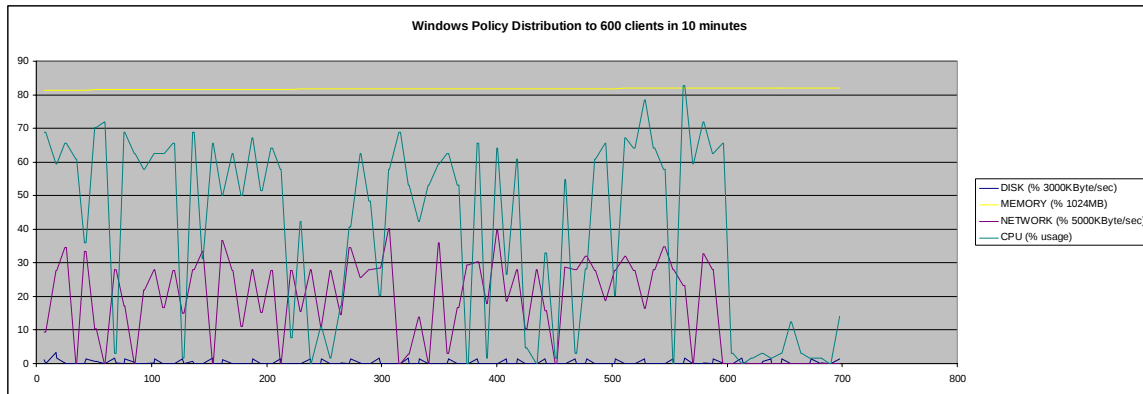
Policy Distribution Load AIX



The policy distribution load is driven by the fact that the clients make multiple requests to the server to fulfill each collector needed. A policy can contain dozens of collectors resulting in dozens of request per client. The bulk of high impact policy distribution will generally occur during the initial stages of an SCM deployment. The server is likely to be in a low load environment since the clients need policies in order to begin to generate the messages that will load the server. Thus, it is difficult to recommend a target CPU resource boundary that should be used as limit to policy distribution load. Also, the throttling of the policy distribution load is difficult in that the load is controlled by moving clients into client

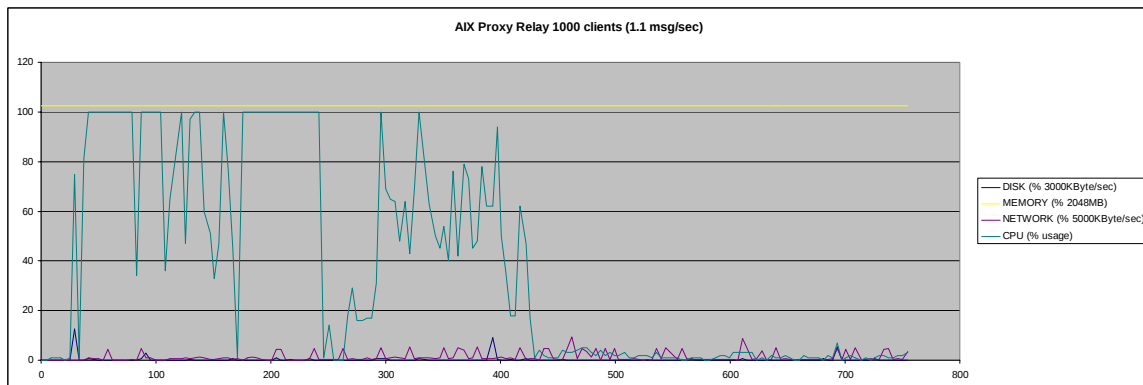
groups or applying policies to client groups that already contain clients. Generally the clients will have to be assigned in stages or the client groups kept small to throttle policy distribution load.

Microsoft Windows



Proxy Impact

AIX

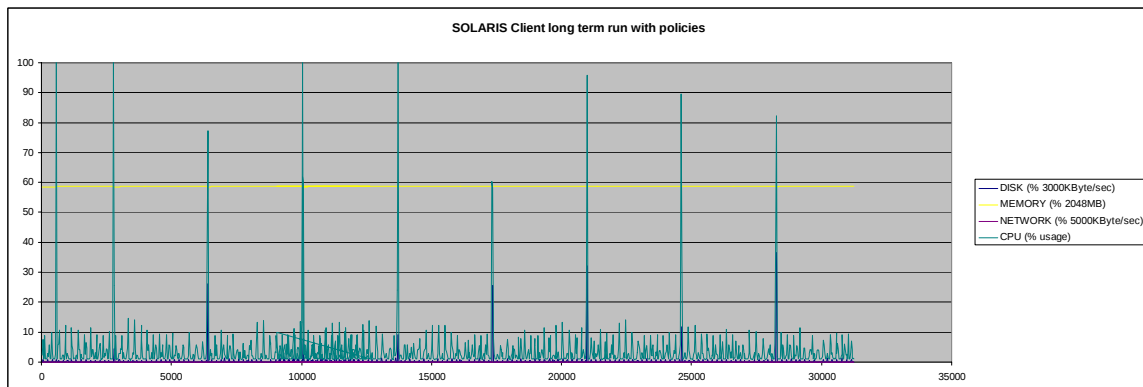
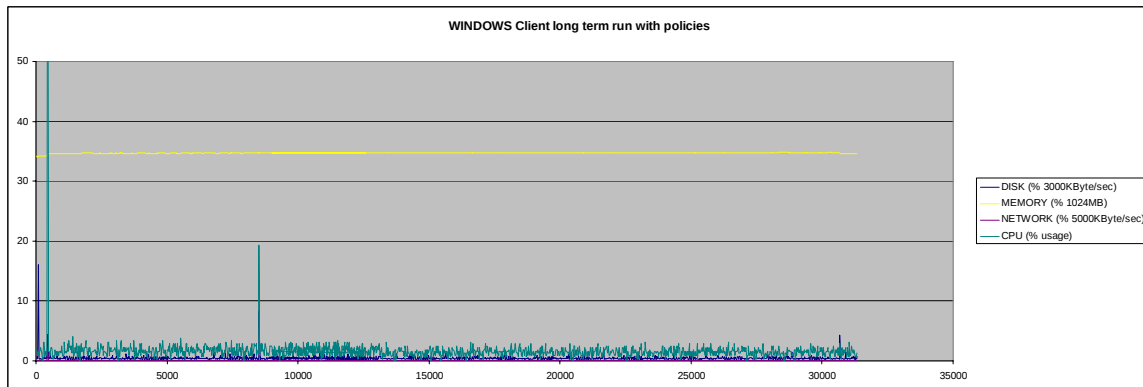
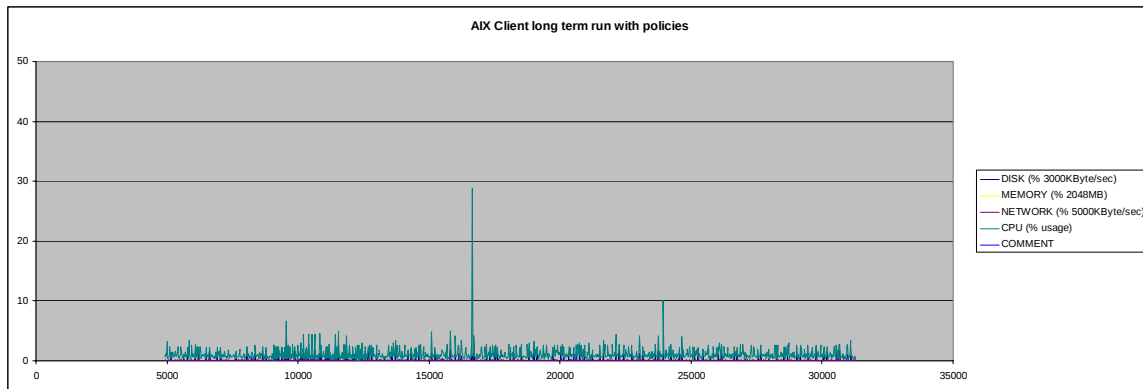


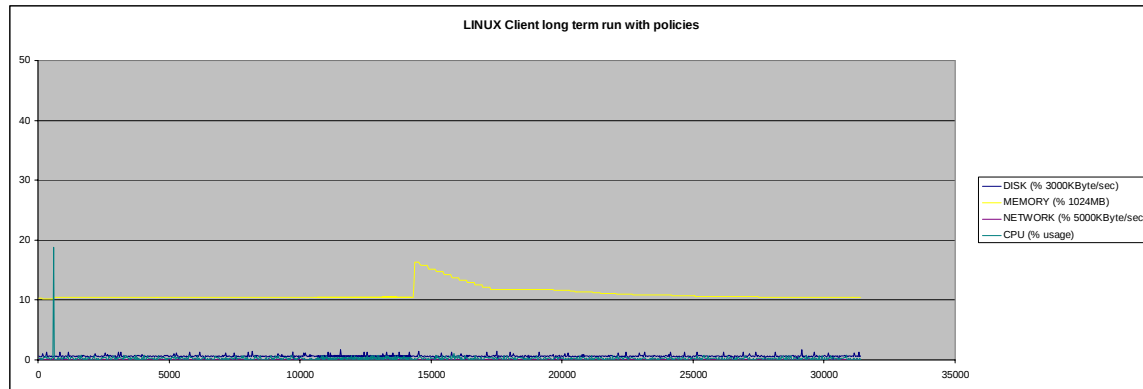
The SCM proxy relay collector produces considerable load only during the server's pull client connection initiation bursts. The server tends to attempt to start dozens of pull client connections simultaneously. These phases overload the proxy relay causing some connection request to time out.

These failed attempts are later retried by the server in the same fashion. Eventually the majority of the pull clients connections are established and the proxy load falls to minimum levels.

During initial testing, out of memory errors occurred after a certain number of client connections were established. This error was caused by the JVM attempting to allocate memory for additional threads. The default setting for the Java thread stacks is 256k, which at 1000 clients and 3 threads per client equals 768MB just for stack space. After reducing the stack setting to 32k, the total memory required for stack space equals 96MB.

Client Long Term Impact





The clients executing proxies exhibited a minimum impact on system regardless of operating system. The spikes that appear on the Solaris graph at regular intervals are being investigated.

CLI Response Times

CLI command	Response Time (sec)
scmregisterclient	5
scmaddgroup	5
scmaddgrouptogroup	4
scmaddgroupclient	26
scmimportpolicy	9
scmexportpolicy	8
scmaddgrouppolicy	4
scmcreatesnapshot	36
scmlistclients (with 10000 clients)	15
scmlistgroups (with 58 groups)	7
scmlistpolicies	4
scmlistusers	4
scmremovegroupclient	27
scmremovegroupfromgroup	5
scmremovegrouppolicy	4
scmremovepolicy	17
scmremovegroup	4
scmunregisterclient	27

On average the CLI commands required a 5 second execution time. This time included invocation, authorizing, and displaying results. Notable exceptions are the snapshot command, which was expected to be time consuming, and the client operations, which were not predicted to deviate from the other commands with respect to response time. This deviation may be due to the total number of clients on the system, which is greater than 10000.

GUI Response Times

GUI operation	Response Time (sec)
---------------	---------------------

GUI startup	8
Logon (10000+ clients)	43
Register client	3
Add group	1
Add client to group (1000 clients)	405
Import Policy	20
Add policy to group	1
Create Snapshot	55
Remove client from group (1000 clients)	205
Remove policy from group	1
Remove policy	15
Remove group	1
Unregister client	1

On average the GUI commands were faster than the equivalent CLI commands due to the fact that each CLI command has to perform an authorization step. The CLI commands were tested as singular operations where one object (client, group, user ...) was affected. On the other hand, and where applicable, GUI operations were performed as bulk operations where many objects were affected at the same time. This testing mirrors the expected customer use, in which CLI operations are used in scripts most likely in loops, and GUI operations are performed on a number of objects.

Hardware Specifications and Recommendations

Hardware Specifications and Recommendations

Machine	CPU Specifications	SPECInt 2000 rating	Hardware multiplier	Maximum recommended throughput (msgs/sec @ 75% CPU usage)
IBM pSeries 610	64-bit POWER3 450Mhz	313	1.0	3.3
IBM pSeries 615	64-bit POWER4 1.2Ghz	739	2.36	7.8
IBM pSeries 630	64-bit POWER4 1.45Ghz	856	2.73	9.0
SUN SunFire V120	UltraSparc Ili 650Mhz	230	1.0	3.3
SUN SunFire V210	UltraSparc IIIi 1Ghz	485	2.11	7.0
SUN SunFire 280R	UltraSparc III Cu 1.2Ghz	637	2.77	9.1
Windows Intel	Intel Pentium 4 2.2Ghz	786	1.0	16.7
Windows Intel	Intel Pentium 4 2.4Ghz	940	1.20	20.0
Windows Intel	Intel Pentium 4 3.0Ghz	1276	1.62	27.0
Linux Intel	Intel Pentium 4 2.2Ghz	786	1.0	20.0
Linux Intel	Intel Pentium 4 2.4Ghz	940	1.20	24.0
Linux Intel	Intel Pentium 4 3.0Ghz	1276	1.62	32.4

Scaling for hardware differences

One method for bridging hardware differences is to use published benchmarks. The Standard Performance Evaluation Corporation (SPEC) provides hardware manufactures a way to publish the results of certain performance benchmarks. Their home web page is at:

<http://www.spec.org>

Since communication programs tend to act like integer arithmetic programs, the benchmark of interest is **specint**. **Specint** results can be found by choosing "SPEC CPU2000" from the home page. Choose "Submitted Results", then SPECint2000. The direct web page for this benchmark is as follows:

<http://www.spec.org/osg/cpu2000/results/cint2000.html>

Look for the machines of interest in any of the submitted results. If a benchmark result lists both machines, those results can be used as a ratio to scale between the two machines. The formula is as follows:

$$\text{<performance of machine one>} = \text{<performance of machine two>} * \text{<specint result for machine one>} / \text{<specint result for machine two>}$$

If there are no benchmark results listing both machines, it may be possible to use an intermediate machine as a bridge. The intermediate machine must be listed in the benchmark results for both of the machines of interest. In this case, ratios can be used to scale from one machine to the intermediate machine and then from the intermediate machine to the second machine.

If a machine does not appear in any benchmark result, it may be similar in architecture to a machine that does appear in the benchmark results. For similar architectures, CPU speed or Mhz ratings are a fairly good factor to use for scaling. In this case, use the ratio of the CPU speeds to scale from one machine to another.

Notices

Copyright International Business Machines Corporation 2004. All rights reserved.
US Government Users Restricted Rights -- Use, duplication or disclosure restricted
by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead.

However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document.

The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication.

IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
224A/101
11400 Burnet Road
Austin, TX 78758
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program describe in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Customers are responsible for ensuring their own compliance with various laws such as the Graham-Leach-Bliley Act, the Sarbanes-Oxley Act, and the Health Insurance Portability and Accountability Act. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal, accounting or auditing advice, or represent or warrant that its products or services will ensure that customer is in compliance with any law.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims

related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
DB2
DB2 Universal Database
IBM
IBM logo
pSeries
RS/6000
Tivoli
Tivoli logo

Microsoft, Windows, Windows NT(R), and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.