RSA ClearTrust 5.0.1
AuthMark Performance Details

By Bruce Weiner
(PDF version, 147 KB)

May 14, 2003

Contents

Disclosure

RSA Security Inc. sponsored the testing in this report. Mindcraft, Inc. conducted the performance tests described in this report at Sun's test lab in Menlo Park, California.

Acknowledgement

We thank Sun for providing the systems used for the tests and the support staff who helped configure the servers.

This part of the white paper analyzes the Login and Extranet Scenario performance of RSA ClearTrust Version 5.0.1.

Login Results Analysis

Table 1 summarizes the Login Scenario performance as a function of the ClearTrust Authorization Server configuration and the number of entries in the directory. All of the Login Scenario tests used one Authorization Server; only the number of CPUs was varied as shown in Table 1.

The RSA ClearTrust Authorization Server is the control point for all authentication and authorization. Our tests were structured to push the Authorization Server system as close as possible to 100% CPU utilization. The *CPU Utilization by Server Type* column in Table 1 shows that the RSA ClearTrust Authorization Server was performing at its maximum. All user credentials for the entries tested were cached in the Authorization Server, which accounts for the low directory server CPU utilization. The Web server CPUs had enough spare performance available so that they were not performance bottlenecks for the tests. The load generator (LG) systems were sufficient to create a workload on the servers that drove the Authorization Server to its peak performance. In Test #3 we did not record the load generator CPU utilization; however, the load generator systems never exceeded 75% CPU utilization for any of the Login tests.

The network had enough bandwidth available to support the highest load without limiting overall performance.

Table 1: RSA ClearTrust 5.0.1 Login Performance

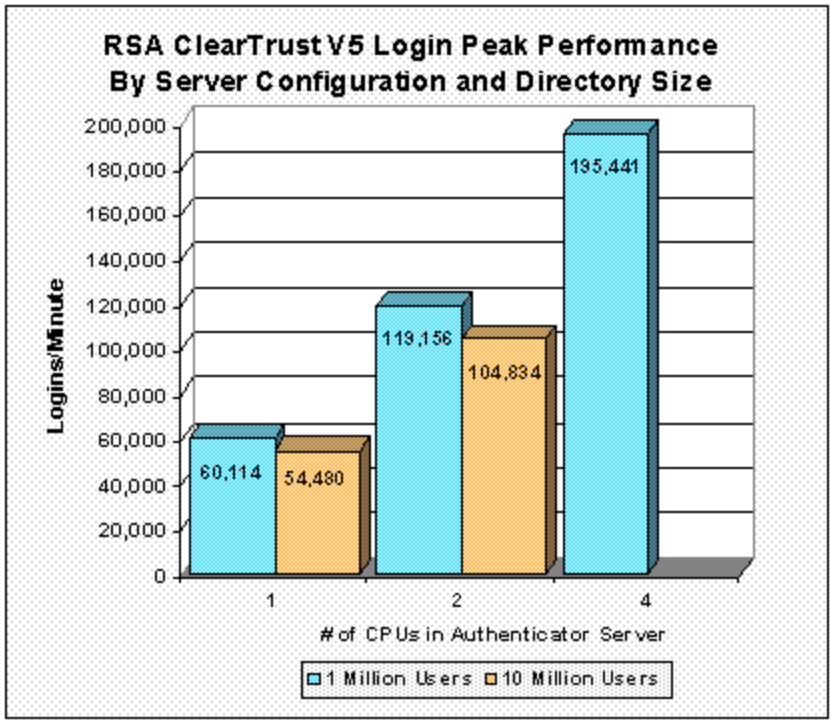
Test ID	Users in Directory	Logins/ Minute	Logins/ Minute/ Authorization Server CPU	Logins/ Minute/ Total CPUs	Scaling Factor/ Total CPUs	Number of CPUs by Server Type	CPU Utilization by System Type
1	1 million	60,114	60,114	6,011	-	Auth.: 1 Web: 8 Directory: 1 Total: 10	Auth.: 97% Web: 75% Directory: 10% LG: 40%
2	1 million	119,156	59,578	6,271	1.04	Auth.: 2 Web: 16 Directory: 1 Total: 19	Auth.: 95% Web: 75% Directory: 10% LG: 75%
3	1 million	195,441	48,860	5,922	0.99	Auth.: 4 Web: 28 Directory: 1 Total: 33	Auth.: 95% Web: 65% Directory: 10% LG: N/A
4	10 million	54,480	54,480	5,448	-	Auth.: 1 Web: 8 Directory: 1 Total: 10	Auth.: 98% Web: 72% Directory: 11% LG: 30%
5	10 million	104,834	52,417	5,518	1.01	Auth.: 2 Web: 16 Directory: 1 Total: 19	Auth.: 97% Web: 70% Directory: 11% LG: 40%

Table 1 also shows normalized metrics based on the total number of CPUs in the configurations tested and the number of CPUs in the single Authorization Server that was used for these tests. The Scaling Factor/Total CPUs in Table 1 shows how much faster on a per-CPU basis a configuration is than an Authorization Server with one CPU, the smallest Authorization Server configuration. It is computed by dividing the Logins/Minute/Total CPUs for a configuration by that for the single-CPU Authorization Server configuration. A metric close to 1.0 indicates linear scaling. A metric above 1.0 means that the configuration provides better than linear scaling while a metric significantly below 1.0 means that the configuration scales less than linearly.

For the 1,000,000-user directory, the Scaling Factor/Total CPUs metric shows that performance scales linearly as the Authorization Server is expanded from one to four CPUs. Keep in mind that for the configurations tested the total number of CPUs did not double from Test #1 to #2 and from Test #2 to #3. The linear performance scaling based on the number of the Authorization Server CPUs is derived from the data in the Logins/Minute/Total CPUs column. For the 10,000,000-user directory, per-CPU Authorization Server performance also scales linearly.

Figure 1 shows RSA ClearTrust's Login performance from Table 1 by Authorization Server configuration and directory size.

Figure 1: RSA ClearTrust Login Performance for 1 Million and 10 Million Users



Extranet Results Analysis

The Extranet Scenario simulates customers or suppliers logging into a private Web site and obtaining information they are authorized to receive. It measures the combination of one user authentication and 10 authorizations for access to resources (these 11 *Extranet operations* constitute one *Extranet sequence*). We report the total Extranet operations per minute, which is 11 times the number of Extranet sequences. The Extranet Scenario, because it uses a more realistic mix of operations than the Login Scenario, provides a better basis for comparing access control and identity management solutions. You can find a more complete description of the Extranet Scenario below.

The Extranet test was done with RSA ClearTrust configured to cache user credentials, as one would do if the servers were on the inside of a private network with firewalls to protect access to systems that store passwords and other sensitive information. We call this a *full-cache* configuration.

Table 2 shows the Extranet Scenario performance of RSA ClearTrust with 1,000,000 users in the directory. As with the Login Scenario, the Extranet Scenario simulates the activities of 10% of the number of users in the directory, which in this case is 100,000 users.

The Extranet Scenario tests were structured to push the Authorization Server system as close as possible to 100% CPU utilization. The *CPU Utilization by Server Type* column in Table 2 shows that the Web servers' CPUs had enough spare performance available so that they were not performance bottlenecks for the tests. At 10% CPU utilization, the directory servers were lightly loaded and did not limit overall performance. Similarly, the network had enough available bandwidth that it did not limit SUT performance.

Table 2 shows that the RSA ClearTrust Authorization Server was performing at its

maximum for Test #6 but still had available CPU time for authentications and authorizations in Test #7. We were not able to push Test #7 to fully utilize the Authorization Server's two CPUs because we did not have enough load generating systems available, which can be seen in their high 83% CPU utilization. Having insufficient computing resources is not uncommon given the exigencies of performance testing. If there had been enough load generating capability, the same SUT would have achieved higher performance.

Table 2: RSA ClearTrust Extranet Performance - 1 Million Users in the Directory

Test ID	Extranet Sequences per Minute	Extranet Operations per minute (authentications + authorizations)	Extranet Operations/ Minute/ Authorization Server CPU	Extranet Operations/ Minute/ Total CPUs	Scaling Factor/ Total CPUs	Number of CPUs by Server Type	CPU Utilization by System Type
6	20,180	221,980 (20,180 + 201,800)	221,980	10,090	-	Auth.: 1 Web: 20 Dir.: 1 Total: 22	Auth.: 98% Web: 60% Directory: 10% LG: 50%
7	35,647	392,117 (35,647 + 356,470)	196,059	12,649	1.25	Auth.: 2 Web: 28 Dir.: 1 Total: 31	Auth.: 85% Web: 75% Directory: 10% LG: 83%

The Scaling Factor/Total CPUs in Table 2 shows how much faster on a per-CPU basis the two-CPU Authorization Server configuration is than the one-CPU configuration. It is computed by dividing the Extranet Operations/Minute/Total CPUs for the two-CPU configuration by that for the one-CPU configuration. This metric clearly shows that performance scales better than linearly as the Authorization Server is expanded from one to two CPUs. This more-than-linear scaling occurred in part because we were able to saturate the Authorization Server in the one-CPU configuration while having excess Web server CPU capacity. In other words, we could have used fewer Web server CPUs and achieved the same level of performance for the one-CPU Authorization Server configuration.

Conclusions

These AuthMark Benchmark results lead us to conclude that:

- RSA ClearTrust 5.0.1 delivers the highest Login performance per Authorization (Policy) Server CPU that we've measured so far, 60,144 Logins/minute/Authorization Server CPU.
- RSA ClearTrust 5.0.1 outperformed all other policy-server-based identity and access management products we tested with the Extranet Scenario both in total Extranet operations per minute (392,117) and in Extranet operations per minute per total CPUs (12,649).
- RSA ClearTrust 5.0.1 delivers outstanding linear performance scaling as Web servers and Authorization Server CPUs are added to an installation.
- RSA ClearTrust 5.0.1 provides excellent, predictable performance for moderately sized communities of 1,000,000 users as well as for large communities of at least 10,000,000 users.

Test Methodology

Mindcraft ® tested the performance of RSA ClearTrust using our iLOAD MVP™ tool to run the AuthMark ™ Benchmark Login and Extranet Scenarios. We describe these tests below to further your understanding of the performance results discussed above.

iLOAD MVP Overview

iLOAD MVP is a general-purpose, script-driven capacity planning, benchmarking, and regression testing tool. The major components of iLOAD MVP are:

- A Control Center that manages client systems, controls test script execution, and reports on test results.
- Multithreaded client load generators that execute test scripts to simulate users accessing a server.
- Test script generation programs.
- Test data generation programs.

iLOAD MVP provides the capabilities needed to test high-performance servers with a small number of client systems. Its capabilities include:

- The ability to simulate a large number of simultaneous user sessions. The number of user sessions is limited only by the client OS, the amount of memory and the performance of the client systems.
- Support for HTTP 1.0 and 1.1 as well as LDAP V3.
- Support for authentication and authorization.
- Support for SSL.
- Custom test scripts.

The AuthMark Benchmark

The AuthMark Benchmark is designed to test the performance of products that provide authentication and authorization services in support of Web servers. *Authentication* is the process of verifying who a user is; it typically occurs when a user logs in. *Authorization* is the process of verifying that an authenticated user is allowed to see or to use a particular resource. In the case of a Web server such resources include HTML files, graphic files, and programs that generate Web pages dynamically.

AuthMark simulates a large number of users accessing Web servers via their browsers. This approach permits AuthMark to test authentication and authorization performance independent of the technology used to provide those services.

AuthMark consists of several test scenarios to determine various aspects of performance for authentication and authorization systems under different circumstances.

AuthMark Login Scenario

The AuthMark Login Scenario focuses on testing authentication. We call it the Login Scenario because authentication is done the first time a user accesses a protected part of a Web site, just like a login. The HTTP 1.0 and 1.1 protocols define the steps a browser follows for authentication. Some of the steps are visible to you and others are not. It is important to understand what happens during a login in order to understand what the Login Scenario measurements mean.

Login Process

The following simplified sequence will walk you through the login process to show how it works using the HTTP 1.0 and 1.1 protocols:

1. When you click on a link or enter a URL in your browser, your browser sends the requested URL to the Web server.

2. The Web server determines that you must be authenticated before it returns the resource at the requested URL. Typically, the authentication requirement is specified as part of the Web server's configuration or via an authentication/authorization product connected to the Web server.

3. The Web server sends back a "401" HTTP response to your browser indicating that you are not authorized to see that requested resource.

4. Your browser pops open a window and asks you to enter your user ID and a password.

5. After you enter your user ID and password, your browser stores them in memory and associates them with the protected space (called a *realm*) containing the URL you requested.

6. Your browser then resends a request for the same URL but this time it includes an HTTP authorization header containing your user ID and password.

7. The Web server checks your user ID and password to see if they match the authentication information in the authentication system. If they do, you are authenticated.

8. Now that you have been authenticated, the authorization system checks whether or not you are authorized to access Web pages in the realm requested. If you are authorized, the Web server sends the Web page you requested.

Notice that the URL you clicked on or entered is actually sent twice (in steps 1 and 6). This means that the authentication system is used twice—first, it finds out that the requested URL requires the user be authenticated, then it processes the authorization header when the request is resent.

Once a user has been authenticated, a Web browser automatically sends the authorization header whenever the user requests a URL in the same realm requiring authentication.

Login Scenario Configuration

Table 3 shows the AuthMark Login Scenario configuration parameters we used.

Table 3: AuthMark Login Scenario Configuration Parameters

Parameter	Values	
Number of users in the directory	1,000,000	10,000,000
Number of Organizational Units or security groups	10	100
Total number of user sessions per test	100,000	1,000,000

The number of user sessions active during a given test run is determined by the length of the test and the number of logins. Sessions are not logged out once created. Instead, each session remains quiescent after login.

Running the Login Scenario

The basic steps for running the Login Scenario are:

A. Generate the data to fill the security database. iLOAD MVP provides a tool to generate realistic data for the LDAP V3 organizationalPerson object class and Netscape's inetOrgPerson object class. It also includes tools to load the same data into an LDAP directory, which was used for this test.

B. Load the security database with the user data.

C. Generate the test scripts for the Login Scenario. iLOAD MVP provides a tool to do this. These scripts drive iLOAD MVP to simulate user interaction with the Web server(s).

D. Load Web pages on the Web server(s). There are 100 Web pages each of which is 14 KB in size for the Login Scenario.

E. Load and configure the user management system or authentication/ authorization system.

F. Run the benchmark.

The Login Scenario test script selects users randomly from those in the directory (see Table 3 for the numbers we used for this test). The tester is free to select the number of load generator systems and the number of iLOAD MVP client threads to use. These are called the *load generators*.

The tester selects the number of load generators to get the highest performance possible from the authentication/authorization system being tested. In order to obtain peak performance, the tester may need to use multiple Web servers and directory servers.

The tester is permitted, but not required, to do a warm-up run of the test scenario in order to get the servers to a state that would more likely represent the state they would be in during normal operation. For this benchmark, we warmed-up the servers by running each test script in its entirety.

Extranet Scenario

The Extranet Scenario simulates an environment where users must login to a Web site and where all access requests require authorization. This scenario depicts a more

realistic usage pattern than the Login Scenario.

The Extranet Scenario test execution starts with the same operation sequence as the Login Scenario (steps 1 - 8 above) and continues with the following operations:

9. The test client requests another resource.
10. The authorization services check the validity of the user and and that the user is authorized to have access to the resource.
11. If the user is authorized, the resource is returned.
12. The test client then requests additional resources.

The Extranet Scenario sequence consists of a total of 11 client operations per user session:

- The first resource request (which does two operations): the login request and the authorization for resource requested; and
- Nine more resource requests that require authorization.

RDA ClearTrus tchecks the continuing validity of an authenticated user each time a resource access request is made to ensure that the user session has not been revoked. However, the user is not re-authenticated. As a result, the user does not see a new login request as long as the resources being accessed are in the Internet domain in which the user has been authenticated.

For the Extranet Scenario, we warmed-up the servers by running the test script in its entirety.

Extranet Scenario Configuration

Table 4 shows the AuthMark Extranet Scenario configuration parameters we used.

Table 4: AuthMark Extranet Scenario Configuration Parameters

Parameter	Values
Number of users in the directory	1,000,000
Number of OrganizationalUnits or security groups	10
Total number of user sessions per test	100,000

Sessions are not logged out once created. Instead, each session remains quiescent after all of its requests have been satisfied.

Running the Extranet Scenario

The basic steps for running the Extranet Scenario are:

1. Generate the data to fill the directory. iLOAD MVP provides a tool to generate realistic data for the LDAP V3 organizationalPerson object class and Netscape's inetOrgPerson object class. It generates the data in standard LDIF format so that it can be loaded directly into an LDAP directory.

2. Load the directory with the user data.

3. Generate the test scripts for the Extranet Scenario. iLOAD MVP provides a tool to do this. These scripts drive the iLOAD clients to simulate user interactions with the Web servers.

4. Load Web pages on the Web servers. There are 100 Web pages each of which is 14 KB in size.

5. Load and configure the authentication/authorization system.

6. Run the benchmark.

The Extranet Scenario test scripts select users randomly from users in the directory (see Table 4 for the numbers we used for these tests). The tester is free to use any number of load generator systems each running any number of iLOAD MVP client threads (called *load generators*).

The tester selects the number of load generators to get the highest performance possible from the authentication/authorization system being tested. In order to obtain the peak performance, the tester may need to use multiple Web servers and directory servers.

The tester is permitted, but not required, to do a warm-up run of the test scenario in order to get the servers in the state they would be in during normal operation. For this benchmark, we warmed-up the servers by running the test scripts in their entirety.

Server Hardware and Software Set Up

Mindcraft used three types of servers for these tests: Web, Authorization, and directory servers. Table 5 shows the summary of the server configurations and the number of load generators we used for each test. The detailed configurations of the RSA ClearTrust Authorization Server and the directory server are shown in Table 6 and Table 7, respectively. Tables 8 and 9 show the Web server configurations. Figure 2 shows how the various servers were connected for the tests; however, you will need to refer to Table 5 to see which systems were used for each test.

Table 5: Servers and Load Generators

Test ID	Web Servers & CPUs	RSA ClearTrust Authorization Server & CPUs	Directory Server & CPUs	Load Generators
Login 1	2 x Sun Fire V480 4 CPUs each	1 x Sun Fire V480 1 CPU	1 x Sun Fire V880 1 CPU	8 x IBM ThinkPad
Login 2	4 x Sun Fire V480 4 CPUs each	1 x Sun Fire V480 2 CPUs	1 x Sun Fire V880 1 CPU	8 x IBM ThinkPad
Login 3	5 x Sun Fire V480 4 CPUs each and 2 x Sun Fire V880 4 CPUs each	1 x Sun Fire V480 4 CPUs	1 x Sun Fire V880 1 CPU	14 x IBM ThinkPad
Login 4	2 x Sun Fire V480 4 CPUs each	1 x Sun Fire V480 1 CPU	1 x Sun Fire V880 1 CPU	8 x IBM ThinkPad
Login 5	4 x Sun Fire V480 4 CPUs each	1 x Sun Fire V480 2 CPUs	1 x Sun Fire V880 1 CPU	14 x IBM ThinkPad
Extranet 6	5 x Sun Fire V480 4 CPUs each	1 x Sun Fire V480 1 CPU	1 x Sun Fire V880 1 CPU	14 x IBM ThinkPad
Extranet 7	5 x Sun Fire V480 4 CPUs each and 2 x Sun Fire V880 4 CPUs each	1 x Sun Fire V480 2 CPUs	1 x Sun Fire V880 1 CPU	14 x IBM ThinkPad

Table 6: RSA ClearTrust Authorization Server Configuration - Sun Fire V480

Feature	Configuration
CPU	4 x 900MHz UltraSPARC III Cu CPUs (we used the `psradm` to enable/disable processors as specified in Table 5) Cache: L1: 32 KB I + 64 KB D; L2: 8 MB (I+D) off chip
RAM	16 GB ECC
Disk	2 x Sun 36GB (Seagate ST336607FSUN36G), FC-AL disk controller, 10,000 RPM
Networks	1 x 1000Base-TX, embedded
Operating System	Solaris 9 with the latest patches
Other Software and Tunes	**Solaris tunes:** ■ None **Java tunes:** ■ -d64 ■ -Xms15g ■ -Xmx15g ■ -XX:NewSize=5g ■ -Xmn5g ■ -XX:SurvivorRatio=100 ■ -XX:MaxTenuringThreshold=0 ■ -XX:PermSize=64m ■ -XX:MaxPermSize=256m ■ -XX:+UseThreadPriorities ■ -XX:NewRatio=2 ■ -XX:+DisableExplicitGC ■ -Xnoclassgc ■ -XX:+UseConcMarkSweepGC **RSA ClearTrust V5.0.1 tunes:** ■ cleartrust.aserver.cache.time_to_live=900 ■ cleartrust.aserver.cache.result=10000000 ■ cleartrust.aserver.cache.url=500000 ■ cleartrust.aserver.cache.url.protection=500000 ■ cleartrust.aserver.cache.webserver=100000 ■ cleartrust.aserver.cache.webserver.protection=500000 ■ cleartrust.aserver.cache.application_function=10000 ■ cleartrust.aserver.cache.entity=10000000 ■ cleartrust.aserver.cache.entitlement=100000000 ■ cleartrust.aserver.cache.smart_rule=500000 ■ cleartrust.aserver.cache.property_definition=1000 ■ cleartrust.aserver.cache.application=1000 ■ cleartrust.aserver.cache.bootstrap.preload=false ■ cleartrust.aserver.cache.bootstrap.preload.log_recommended=true ■ cleartrust.aserver.thread_pool_size=28 ■ cleartrust.aserver.max_thread_pool_size=256 ■ cleartrust.aserver.max_connections=256 ■ cleartrust.runtime_api.socket.tcp_nodelay=false ■ cleartrust.aserver.lease_renewal=13000 ■ cleartrust.aserver.authorization_mode=active

Table 7: Directory Server Configuration - Sun Fire V880

Feature	Configuration
CPU	4 x 900MHz UltraSPARC III Cu CPUs (we used the `psradm` to enable a single processor) Cache: L1: 32 KB I + 64 KB D; L2: 8 MB (I+D) off chip
RAM	32 GB ECC
Disk	5 x Sun 72GB (Seagate ST373307FSUN72G), FC-AL disk controller, 10,000 RPM
Networks	1 x 1000Base-TX, embedded
Software	Solaris 9 with the latest patches
Other Software and Tunes	**Solaris tunes:** ■ In /etc/system: ■ set shmsys:shminfo_shmmax=0xffffffff ■ set shmsys:shminfo_shmseg=32 **Sun ONE Directory Server 5.2 tunes:** ■ None

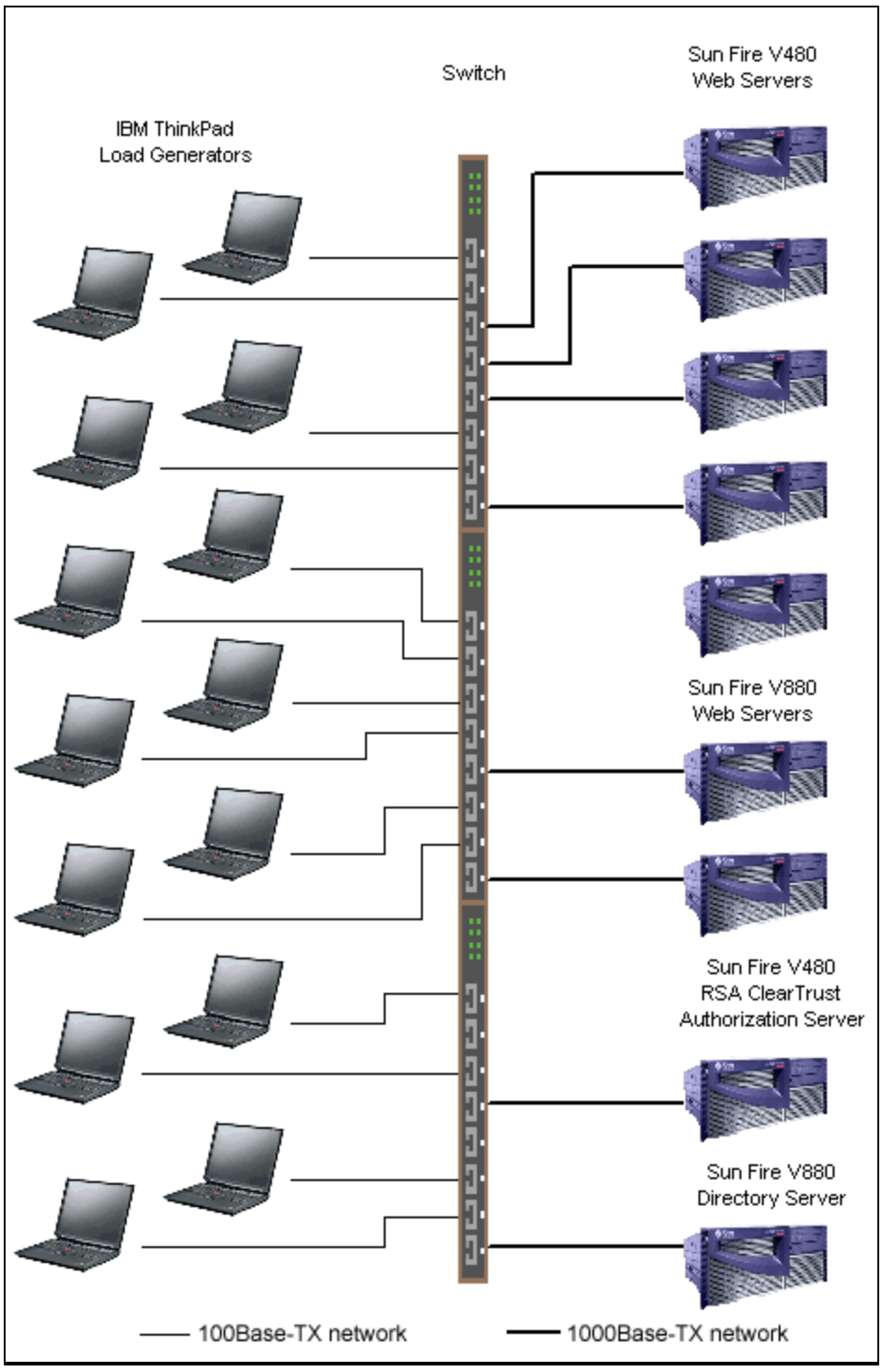
Table 8: Web Server A Configuration - Sun Fire V480

Feature	Configuration
CPU	4 x 900MHz UltraSPARC III Cu CPUs Cache: L1: 32 KB I + 64 KB D; L2: 8 MB (I+D) off chip
RAM	16 GB ECC
Disk	2 x Sun 36GB (Seagate ST336607FSUN36G), FC-AL disk controller, 10,000 RPM
Networks	1 x 1000Base-TX, embedded
Operating System	Solaris 9 with the latest patches
Other Software and Tunes	**Solaris tunes:** ■ None **Sun ONE Web Server 6.0, Service Pack 5 tunes:** ■ Web server tunings (magnus.conf): ■ RqThrottle 2048 ■ ListenQ 256 ■ UseNativePoll on ■ KeepAliveThreads 8 ■ MaxKeepAliveConnections 8192 ■ RcvBufSize 8192 ■ ernelThreads on ■ StrictHttpHeaders off ■ PostThreadsEarly on ■ KeepAliveTimeout 3600 ■ IOTimeout 10 ■ RqThrottleMin 512 ■ KeepAliveQueryMeanTime 50 ■ KeepAliveQueryMaxSleepTime 70 ■ Servlets and JSPs disabled ■ Web server tunings (oj.conf): ■ CGIs, and directory indexing disabled ■ Web server tunings (server.xml): ■ acceptorthreads="4" **RSA ClearTrust Agent, version 4, SunONE 6 tunes:** ■ cleartrust.agent.idle_timeout=95 Mins (default=15 Mins) ■ cleartrust.agent.cookie_ip_check=False (default=True) ■ cleartrust.agent.cookie_touch_window=30 Mins (default=30 Secs) ■ cleartrust.agent.export_session_init_time=False (default=True) ■ cleartrust.agent.export_session_expiration_time=False (default=True) ■ cleartrust.agent.export_last_touch_time=False (default=True) ■ cleartrust.agent.export_cookie_user_buffer=False (default=True) ■ cleartrust.agent.form_based_enabled=False (default=True) ■ cleartrust.agent.retain_uri=False (default=True) ■ cleartrust.agent.protected_resource_cache_ttl=90 Mins (default=10 Mins) ■ cleartrust.agent.protected_resource_cache_size=1000000 (default=100000) ■ cleartrust.agent.unprotected_resource_cache_ttl=90 Mins (default=5 Mins) ■ cleartrust.agent.unprotected_resource_cache_size=1000000 (default=10000) ■ cleartrust.agent.authz_allow_cache_ttl=90 Mins (default=5 Mins) ■ cleartrust.agent.authz_allow_cache_size=1000000 (default=10000) ■ cleartrust.agent.authz_deny_cache_ttl=90 Mins (default=10 Mins) ■ cleartrust.agent.authz_deny_cache_size=1000000 (default=10000) ■ cleartrust.agent.token_cache_ttl=90 Mins (default=5 Mins ■ cleartrust.agent.token_cache_size=1000000 (default=10000) ■ cleartrust.agent.user_properties_cache_ttl=90 Mins (default=10 Mins) ■ cleartrust.agent.dispatcher_timeout=90 Secs (default=10 Secs) ■ cleartrust.agent.ssl.use=Clear (default=Anon)

Table 9: Web Server B Configuration -Sun Fire V880

Feature	Configuration
CPU	4 x 900MHz UltraSPARC III Cu CPUs Cache: L1: 32 KB I + 64 KB D; L2: 8 MB (I+D) off chip
RAM	32 GB ECC
Disk	5 x Sun 72GB (Seagate ST373307FSUN72G), FC-AL disk controller, 10,000 RPM
Networks	1 x 1000Base-TX, embedded
Operating System	Solaris 9 with the latest patches
Other Software and Tunes	**Solaris tunes:** ■ None **Sun ONE Web Server 6.0, Service Pack 5 tunes:** ■ Web server tunings (magnus.conf): ■ RqThrottle 2048 ■ ListenQ 256 ■ UseNativePoll on ■ KeepAliveThreads 8 ■ MaxKeepAliveConnections 8192 ■ RcvBufSize 8192 ■ ernelThreads on ■ StrictHttpHeaders off ■ PostThreadsEarly on ■ KeepAliveTimeout 3600 ■ IOTimeout 10 ■ RqThrottleMin 512 ■ KeepAliveQueryMeanTime 50 ■ KeepAliveQueryMaxSleepTime 70 ■ Servlets and JSPs disabled ■ Web server tunings (oj.conf): ■ CGIs, and directory indexing disabled ■ Web server tunings (server.xml): ■ acceptorthreads="4" **RSA ClearTrust Agent, version 4, SunONE 6 tunes:** ■ cleartrust.agent.idle_timeout=95 Mins (default=15 Mins) ■ cleartrust.agent.cookie_ip_check=False (default=True) ■ cleartrust.agent.cookie_touch_window=30 Mins (default=30 Secs) ■ cleartrust.agent.export_session_init_time=False (default=True) ■ cleartrust.agent.export_session_expiration_time=False (default=True) ■ cleartrust.agent.export_last_touch_time=False (default=True) ■ cleartrust.agent.export_cookie_user_buffer=False (default=True) ■ cleartrust.agent.form_based_enabled=False (default=True) ■ cleartrust.agent.retain_uri=False (default=True) ■ cleartrust.agent.protected_resource_cache_ttl=90 Mins (default=10 Mins) ■ cleartrust.agent.protected_resource_cache_size=1000000 (default=100000) ■ cleartrust.agent.unprotected_resource_cache_ttl=90 Mins (default=5 Mins) ■ cleartrust.agent.unprotected_resource_cache_size=1000000 (default=10000) ■ cleartrust.agent.authz_allow_cache_ttl=90 Mins (default=5 Mins) ■ cleartrust.agent.authz_allow_cache_size=1000000 (default=10000) ■ cleartrust.agent.authz_deny_cache_ttl=90 Mins (default=10 Mins) ■ cleartrust.agent.authz_deny_cache_size=1000000 (default=10000) ■ cleartrust.agent.token_cache_ttl=90 Mins (default=5 Mins ■ cleartrust.agent.token_cache_size=1000000 (default=10000) ■ cleartrust.agent.user_properties_cache_ttl=90 Mins (default=10 Mins) ■ cleartrust.agent.dispatcher_timeout=90 Secs (default=10 Secs) ■ cleartrust.agent.ssl.use=Clear (default=Anon)

Figure 2: Test Lab Configuration
(Some servers and load generators are not used for all tests. See Table 5 for details)



Load Generator Systems

We used up to 14 IBM ThinkPad laptop computers as load generator systems for these tests. Table 10 details the configuration of the load generator systems.

Table 10: IBM ThinkPad Configuration

Feature	Configuration
CPU	1 x Intel® Pentium® III 500 MHz CPU Cache: L1: 16 KB I + 16 KB D; L2: 256 KB (I+D)
RAM	320 MB RDRAM
Disk	1 x 18.6 GB Toshiba MK20169GAP IDE disk
Networks	1 x Intel PRO/100+ MiniPCI
Operating System	Microsoft Windows 2000 Advanced Server, SP2
Tunes	**Windows 2000 Advanced Server tunes:** ■ HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\MaxUserPort=EA60

For more information, contact us at: info@mindcraft.com
Phone: +1 (408) 395-2404
Fax: +1 (408) 395-6324