385 South 520 West
Lindon, UT 84042
(801) 226-8200
Fax: (801) 226-8205
Email: info@keylabs.com
URL: http://www.keylabs.com

Comparison Report

Microsoft ACTIVE DIRECTORY
vs.
NetWare NDS

February 2000

TABLE OF CONTENTS

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

LIST OF FIGURES

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

1.0 PERFORMANCE SUMMARY

KeyLabs, a leading performance testing lab, compared the Lightweight Directory Access Protocol (LDAP) performance of Microsoft Active Directory on Windows 2000 Server to Novell NDS 8 on NetWare 5.1. The test was commissioned by Microsoft Corp.

1.1 Testing Methodology

The tests were designed to show how each directory performs and scales in an LDAP environment. Each directory was tested under four directory sizes ranging form 10,000 objects to 5 million objects. Using the Netscape LDAP API, custom PERL scripts were executed from four Sun Microsystems servers to generate the client load. Finally, four different LDAP search tests were performed:

- Base Object Non-Randomized.
- Base Object Randomized
- Sub-Tree Search Randomized
- Base Object Non-Randomized, 1 Attribute

All tests were run 3 times, and the recorded result is an average. The Directories and associated NOS were installed using default parameters except as noted later in the report. Tests were run on identical hardware and using identical resources to conduct and record the test results. After the three runs were recorded, the server was restarted to clean up cache and free memory for the next test. In addition, to demonstrate how each platform scales on multiprocessor servers, each directory was testing on a single processor configuration and a four-processor server configuration. In total, there were 96 different tests.

1.2 Results Summary

The test data support the following conclusions:

Active Directory significantly outperformed NDS under all tests. In all of the 96 different tests, Active Directory performed significantly faster, in some cases up to 5-times faster, than NDS 8 running on NetWare 5.1.

Active Directory provided better SMP scaling. While the LDAP performance of NDS 8 on NetWare 5.1 didn't show significant increases in throughput when tested on the four-processor configuration, the LDAP performance of Active Directory improved significantly. The results also show that the LDAP performance of Active Directory on the single processor configuration is significantly better than the LDAP performance of NDS on the four-processor server.

Both Active Directory and NDS maintained the LDAP throughput as the number of objects increased. When comparing throughput numbers that each directory achieved when running the 10,000 object test to the throughput numbers achieved when running

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

the 5 million object test, the LDAP throughput of Active Directory deviated only 15 percent on the single processor configuration and only 10 percent on the four-processor configuration. Similarly, the LDAP throughput of NDS 8 deviated only 24 percent on the single processor configuration and only 12 percent on the four-processor configuration.

2.0 TEST METHODOLOGY

2.1 Hardware Configuration

The testing was conducted on the following hardware configurations for both NetWare NDS and Microsoft Active Directory. Performance scripts were run on the Sun Microsystems box(s) for both configurations.

Tests were run on a Compaq Proliant 8000. The server consisted of 1 or 4 Intel Pentium III 550Mhz Xeon processors with 1MB of L2 cache, 2GB of RAM, Intel Pro 100b 10/100Mb PCI NIC and a Compaq Smart Array 3100ES RAID controller. The RAID array consisted of seven 9GB 10,000-rpm LVD SCSI drives and was optimized for read access.

Test scripts were run from 1 or more SUN UE450 systems with four 333Mhz processors, 1GB RAM, 18GB hard drive and a 100Mb NIC. When running tests on the Compaq server with 1 processor, the scripts were run on identical hardware listed above, except the system had two 333Mhz processors instead of four.

All testing was done using 100mb Ethernet with a Cisco Catalyst 2900 XL switch. Networks were isolated and there was no traffic on the wire except from the targeted server and test client(s).

2.1.1 NetWare

NetWare 5.1 was installed from shipping source using default installation parameters, except as noted in Section 4.1 in this report. The latest e-directory evaluation was downloaded from Novell's site and applied.

2.1.2 Microsoft Active Directory

Windows 2000 Server (Build 2194) was supplied by Microsoft. KeyLabs used the default installation parameters, except as noted in Section 4.2. Active Directory Server was then installed using default installation parameters, except as noted in Section 4.2.

2.2 Performance Script Details

LDIF data was created and loaded to each directory. This data contained only user/contact type objects with limited attributes and values. The UID is not realistic, in that it is quite long. This was done because some of the required testing was done with object counts in the millions. The UID was created this way to avoid duplication and allow a large number of objects. This was acceptable because the overall size of the object would be similar to a real-world user in that it would have a smaller UID, but more attributes.

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Sample from the NetWare LDIF file:

```
# LDIF entries added by parent proc -- 15:04:49 01/06/00
dn: dc=hq
objectclass: top
objectclass: domain
dc: hq

dn: dc=1, dc=hq
objectclass: top
objectclass: domain
dc: 1

# LDIF entries added by child 1 on round 1 -- 15:04:49 01/06/00
dn: cn=7360-923432689-887208-1@1.hq, dc=1, dc=hq
objectclass: organizationalperson
cn: Fs Trrb
sn: Trrb
mail: 7360-923432689-887208-1@1.com
title: n7360 923432689
employeenumber: 7360-923432689-887208-1
description: entry of user 7360-923432689-887208-1
telephonenumber: +1 541 625 7302
facsimiletelephonenumber: +1 541 625 8582
ou: n7360
roomnumber: N13
l: Tnooroed
postaladdress: 736 O street $ Tnooroed, OE 73600 $ US
postalcode: 73600
street: 736 O street
```

Sample from the Active Directory LDIF file:

```
# LDIF entries added by parent proc -- 15:04:49 01/06/00

dn: ou=1, dc=hq, dc=kldom, dc=com
objectclass: top
objectclass: organizationalunit
ou: 1

# LDIF entries added by child 1 on round 1 -- 15:04:49 01/06/00
dn: cn=7360-923432689-887208-1@1.com, ou=1, dc=hq, dc=kldom, dc=com
objectclass: contact
cn: Fs Trrb
sn: Trrb
mail: 7360-923432689-887208-1@1.com
title: n7360 923432689
employeenumber: 7360-923432689-887208-1
description: entry of user 7360-923432689-887208-1
telephonenumber: +1 541 625 7302
facsimiletelephonenumber: +1 541 625 8582
ou: n7360
roomnumber: N13
l: Tnooroed
postaladdress: 736 O street $ Tnooroed, OE 73600 $ US
postalcode: 73600

street: 736 O street
```


KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

The different directories required slightly different objects as pertains to their schemas. NDS was defined with Object Class = organizationalperson. AD was defined with Object Class = Contact. The tree structure was designed to ensure that the objects in each directory were the same distance from the root.

Data of this type was considered "seed" data, or data that would be searched on. The 10k, 100k and 1 million object tests were all seed data while the 5 million object test had 3 million objects of seed data and other objects added to bring the total to the target level of 5 million. Both the seed and filler objects were located in multiple OU's.

A database (DB) file was created from this LDIF file for the scripts to use in determining what items to search for. Every object in the LDIF file was also in the associated DB file. This was important in order to ensure realistic searching on all or any object.

In all test cases except for the Base Object Non-Randomized, 1 Attribute tests, the search requests were for each targeted object and 6 attributes. CN, SN, Telephonenumber, Facsimiletelephonenumber, mail, postaladdress. The results were verified against the DB file. In the case of the 1 Attribute test, the only requested attribute was CN.

All testing was done with Non-SSL connection type.

2.2.1 Base Object Search Non-Random

An LDAP base object search is defined as an instance where the client application knows the exact container (organizational unit) where the object resides. For example, if the object that client is searching for resides in the LDAP path cn=west,cn=sales,dc=keylabs,dc=com, then the search will start at this location.

Non-Random refers to how the benchmark client searches for each object. For example, if there are 1,000 named c1, c2, …, c1000, it will search for c1 first, c2 second, …, and c1000 last.

The script utilizes the created DB file to determine the number of objects to be searched and for each specified client. It searches for X number of objects from the start of the DB until the appropriate number of searches is reached.

Example:

> If the specified script was defined for 10 clients, each running 10,000 threads (or virtual clients) for a total of 100,000 searches, then each client would read the first 10,000 objects from the DB file and search sequentially for each object.

This type of test is inherently faster because it allows the target to utilize Cache and reply to the requests with higher performance. All 10 clients are looking for the same 10,000 objects.

2.2.2 Base Object Search Random

Randomized refers to how the benchmark client searches for each object. In this case, each object was queried randomly. For example, if the search was for clients ranging

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

from c1 – c1000, then it searched for these in a random order – c44, c993, c543, c2, … until the desired number of searches had been reached.

This script utilized the created DB file to determine the number of objects to be searched for and for each specified client. It searched for X number of objects RANDOMLY until the appropriate number of searches was reached.

Example:

> If the specified script was for 10 clients, each running 10,000 threads for a total of 100,000 searches, then each client would read the RANDOMLY select and search for 10,000 objects from the DB file.

This type of test is more realistic and better gauges the targets ability to service multiple random search requests. Obviously, there is a chance of some duplication, but in a DIB of millions, the odds are that this would be small. Occasional duplication is also a realistic scenario.

2.2.3 Sub-Tree Search Randomized

A sub-tree search differs form a base search in that the client doesn't know the exact container where the object resides. Using the base search example above, a sub-tree search would start at dc=keylabs,dc=com and search through all the containers located under this root. As the result show, this type of search is more intensive than a base search.

This script utilized the created DB file to determine the number of objects to be searched for and for each specified client. It searched for X number of objects RANDOMLY from the specified base container level, until the appropriate number of searches was reached.

Example:

> The specified script was for 10 clients each running 10,000 threads for a total of 100,000 searches each client would read the DB file and RANDOMLY select 10,000 objects to search for.
>
> For instance, if the specified script was defined for 10 clients, each running 10,000 threads (or virtual clients) for a total of 100,000 searches, then each client would read the first 10,000 objects from the DB file and search RANDOMLY for each object.

This type of test is more intensive and gauges the target's ability to service multiple random search requests across the entire sub-tree structure. Obviously, there is a chance of some duplication, but in a DIB of millions, the odds are that this would be small. Occasional duplication is also a realistic scenario.

2.2.4 Base Object Search Non-Random, 1 Attribute

This script utilized the created DB file to determine the number of objects to be searched for and for each specified client. It searched for X number of objects from the start of the DB, until the specified number of searches was reached. This search asked for only one attribute to be returned, instead of 6.

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Example:

> If the specified script was for 10 clients, each running 10,000 threads for a total of 100,000 searches, then each client would read the first 10,000 objects from the DB file and conduct a search for these objects.

This type of test is inherently faster because it allows the target to utilize Cache and reply to the requests with higher performance. All 10 clients are looking for the same 10,000 objects. It is the fastest search possible, because it returns and verifies only one attribute.

3.0 TEST RESULTS

3.1 Performance Testing – 10K Objects

At 10,000 objects, all 10,000 objects were created using custom PERL scripts. The objects were loaded using Bulkloader on NDS 8, and using LDIFDE on Active Directory. All 10,000 objects were seed data. The tests were run on 1 & 4 processor configurations to illustrate the platforms scalability.

3.1.1 Base Object Non-Randomized

NetWare performed consistently across all tests. Active Directory experienced a dramatic increase while doing the 1500x250 searches. In all tests, Active Directory performed twice as many searches in the same amount of time as NDS. In some cases, it completed seven-times more searches for both single and quad processor.

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Figure 1: BS, 4 Processors

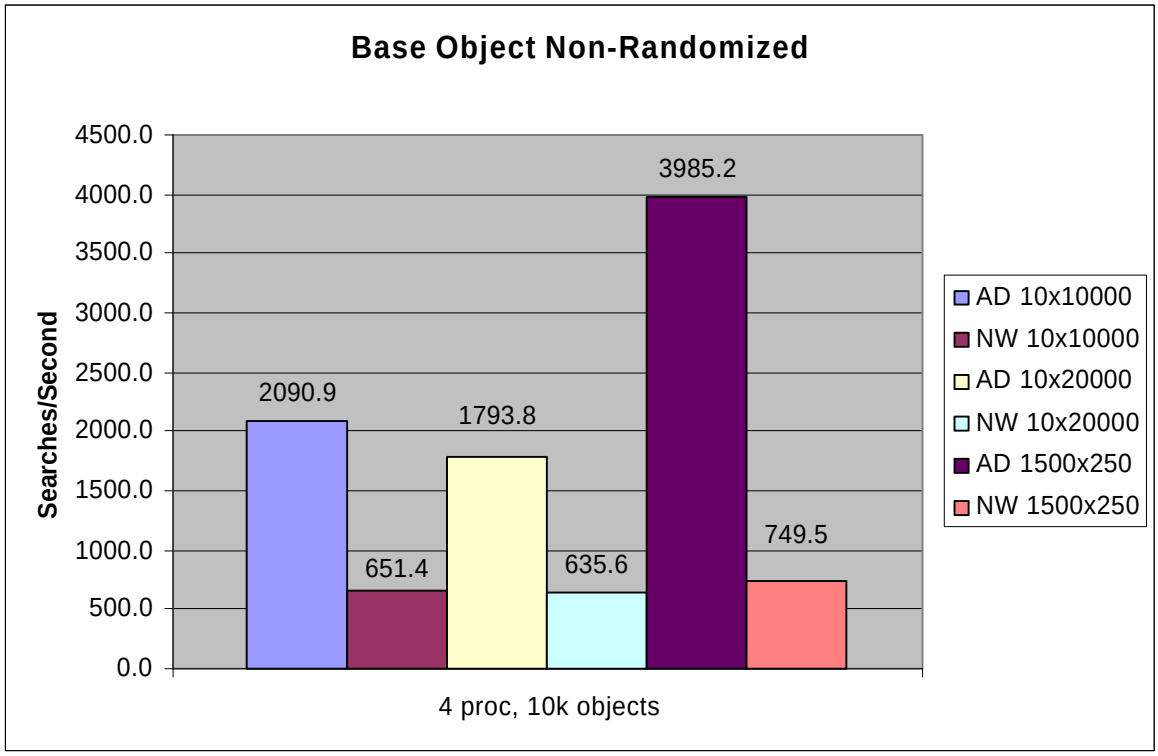
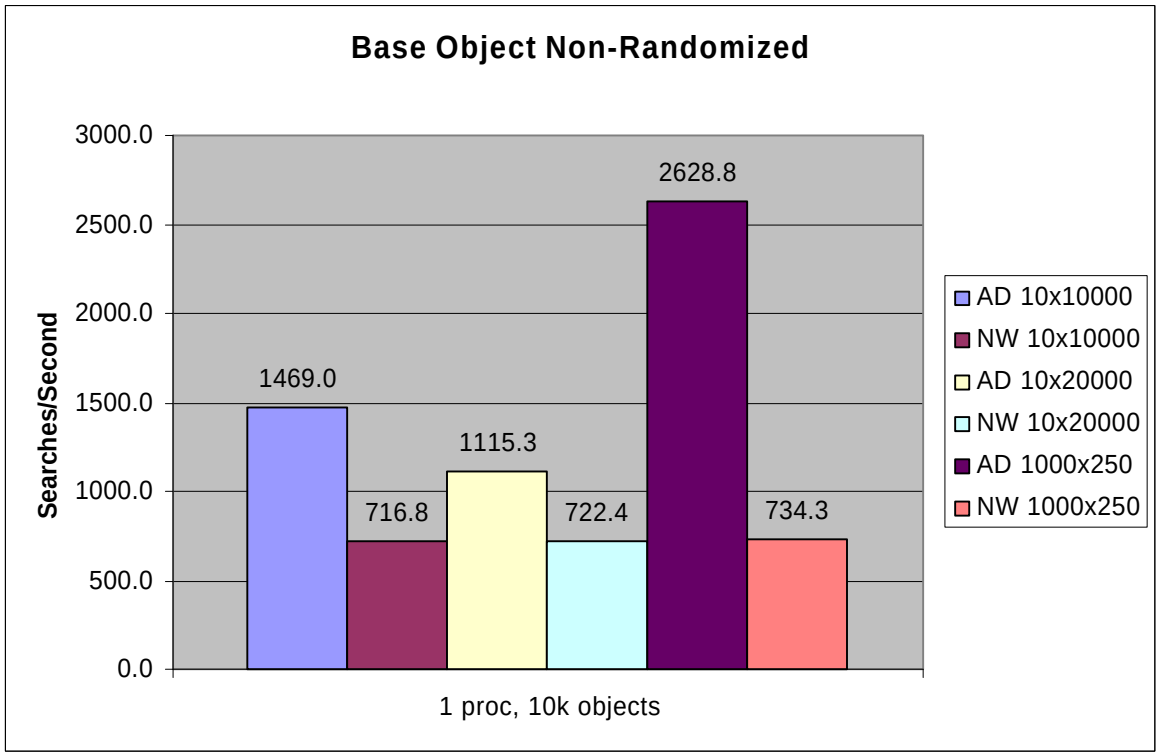


Figure 2: BS, 1 Processor



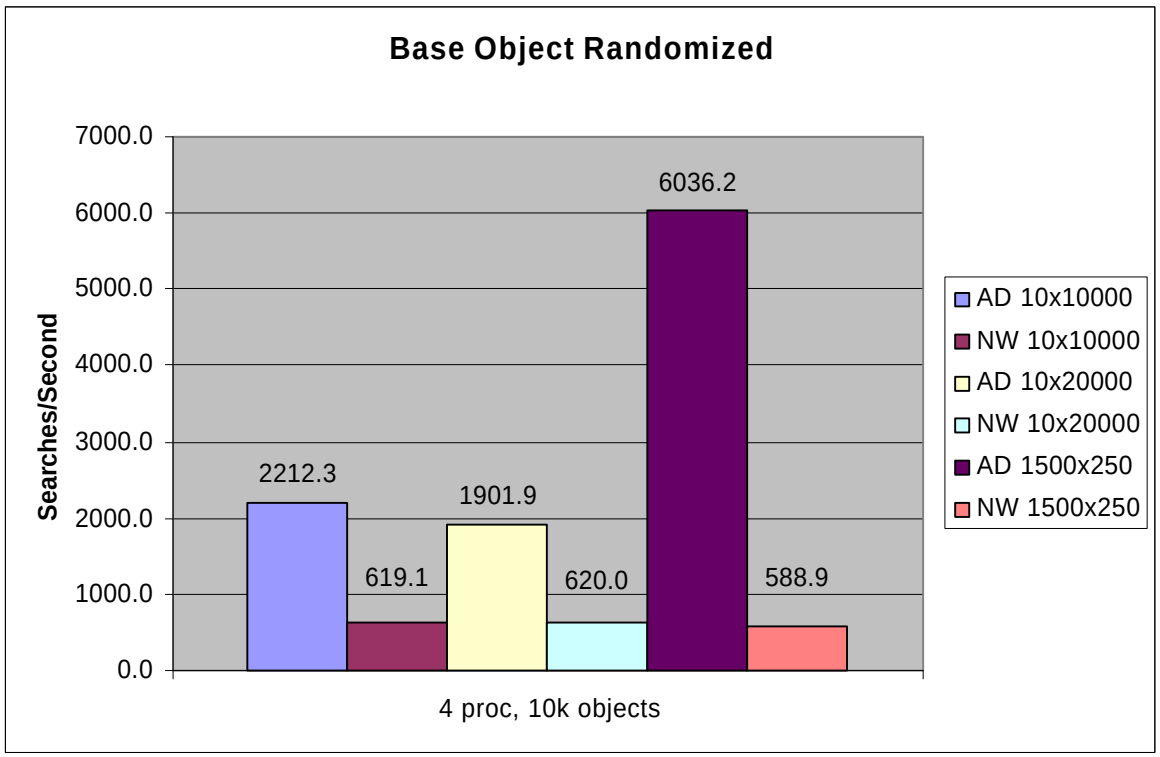
KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Microsoft Windows 2000 with 4 processors was able to provide excellent search times, with 50-92% CPU Utilization. With 1 processor, Active Directory was still producing search results faster than NDS, although by not as wide a margin. NDS was actually a little faster with 1 processor for these tests, than it was with 4 processors. In all cases, regardless of number of processors, the system was above 95% in CPU Utilization.

3.1.2 Base Object Randomized

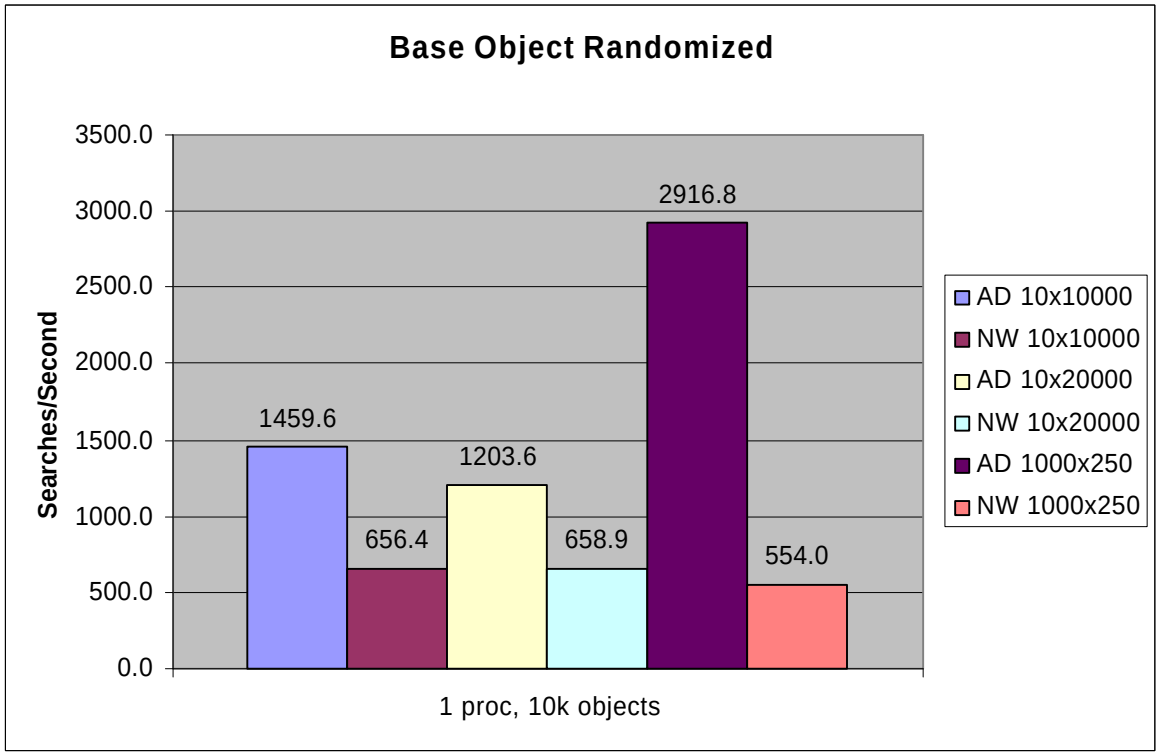
This test is valuable in showing a target's ability to search for random objects. The target is not always able to utilize the cache, but if a good caching model is available the 1500x250 test will expose it, as all 1500 clients are looking for the same 250 random objects. In all cases, Active Directory was able to maintain or beat the results from the non-randomized tests, while NDS experienced a slight drop.

Figure 3: BSD, 4 Processors



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Figure 4: BSD, 1 Processor



3.1.3 Sub-Tree Search Randomized

This test exercised the directory to its capabilities. All searches were random, and must occur across the entire sub-tree structure. Both platforms experienced a drop in search times, but Windows 2000 still maintained a significant performance and scalability advantages over NetWare in all tests while utilizing fewer CPU resources.

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Figure 5: SSD, 4 Processors

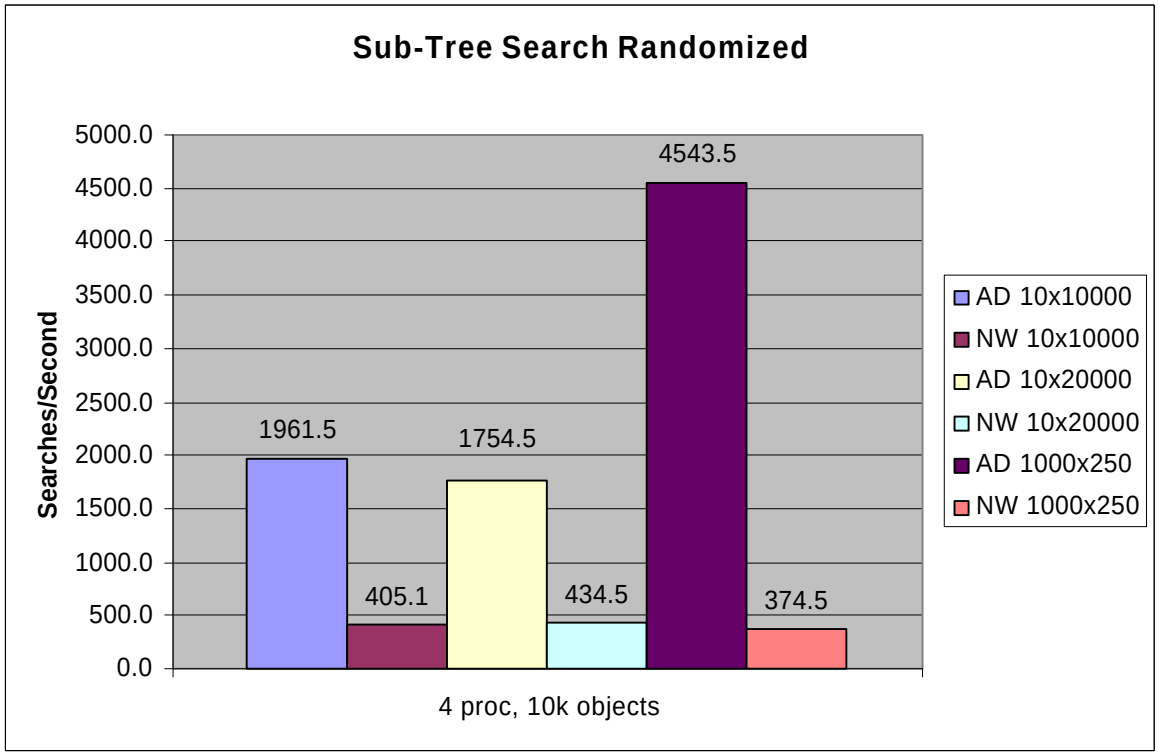
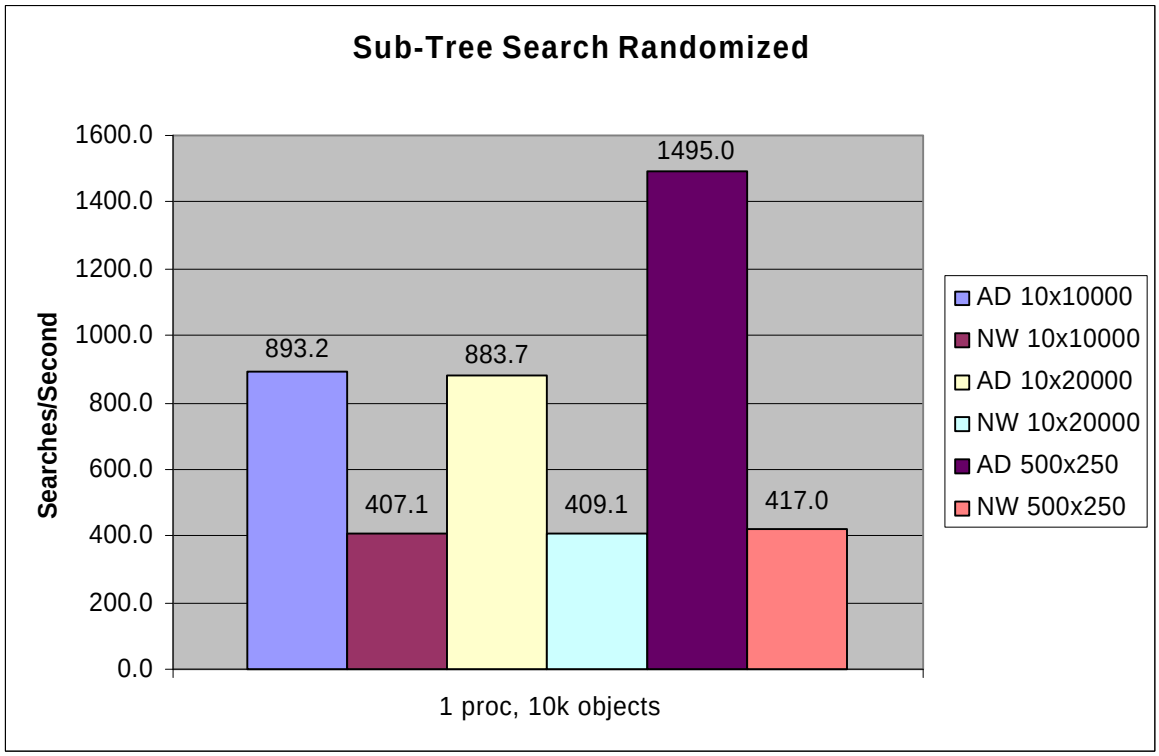


Figure 6: SSD, 1 Processor

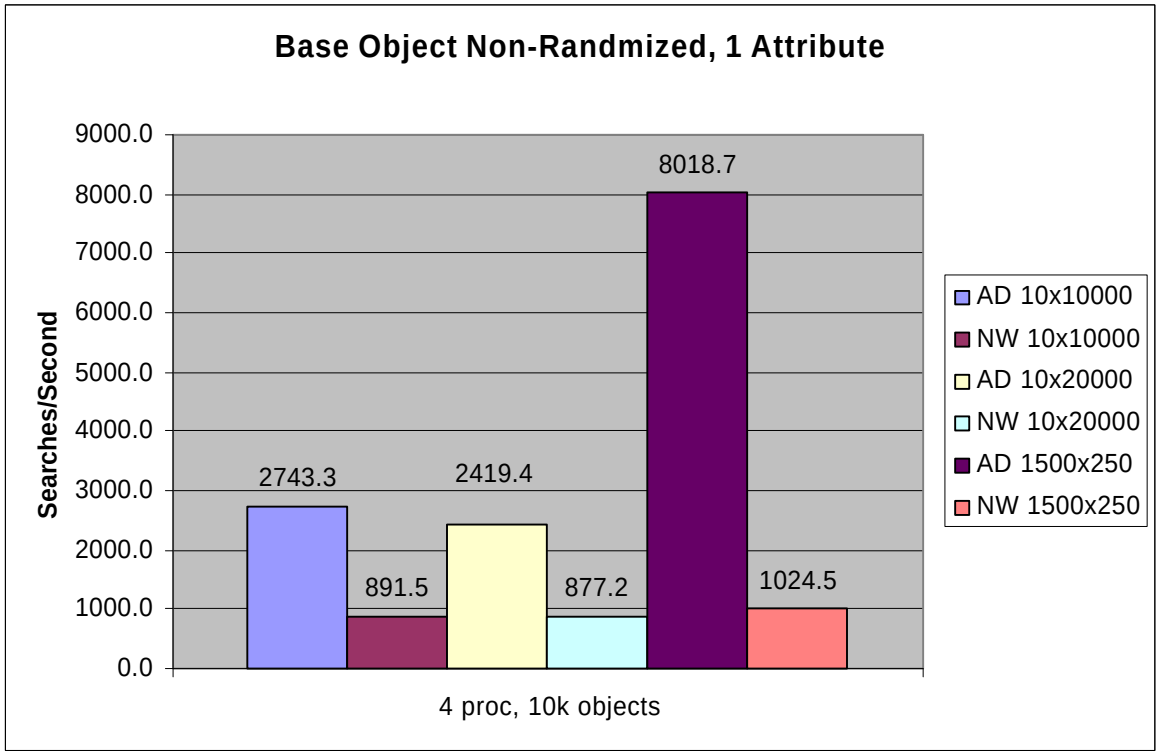


KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.1.4 Base Object Non-Randomized, 1 Attribute

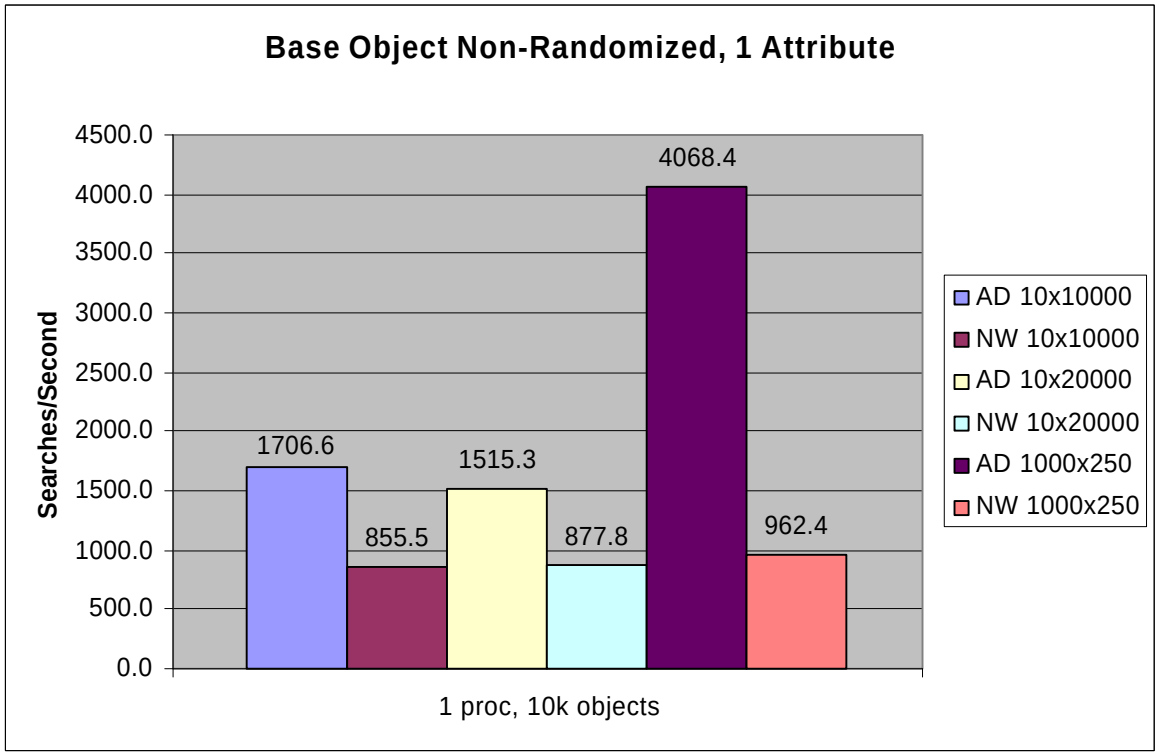
This test was the easiest and fastest test to perform. It illustrated the target's ability to respond to many simple requests and then move on. This test is identical to the Base Object Non-Randomized test, except that instead of responding to requests with 6 attributes, the target is only asked to provide 1 attribute.

Figure 7: BS1A, 4 Processors



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Figure 8: BS1A, 1 Processor



3.2 100k Object Performance Testing

At 100k objects, all objects were created using custom PERL scripts. The objects were loaded using Bulkloader on NDS 8, and LDIFDE on Active Directory. All 100k objects were seed data. The tests were run on 1 & 4 processor configurations to illustrate the platform's scalability.

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.2.1 Base Object Non-Randomized

Figure 9: BS, 4 Processors

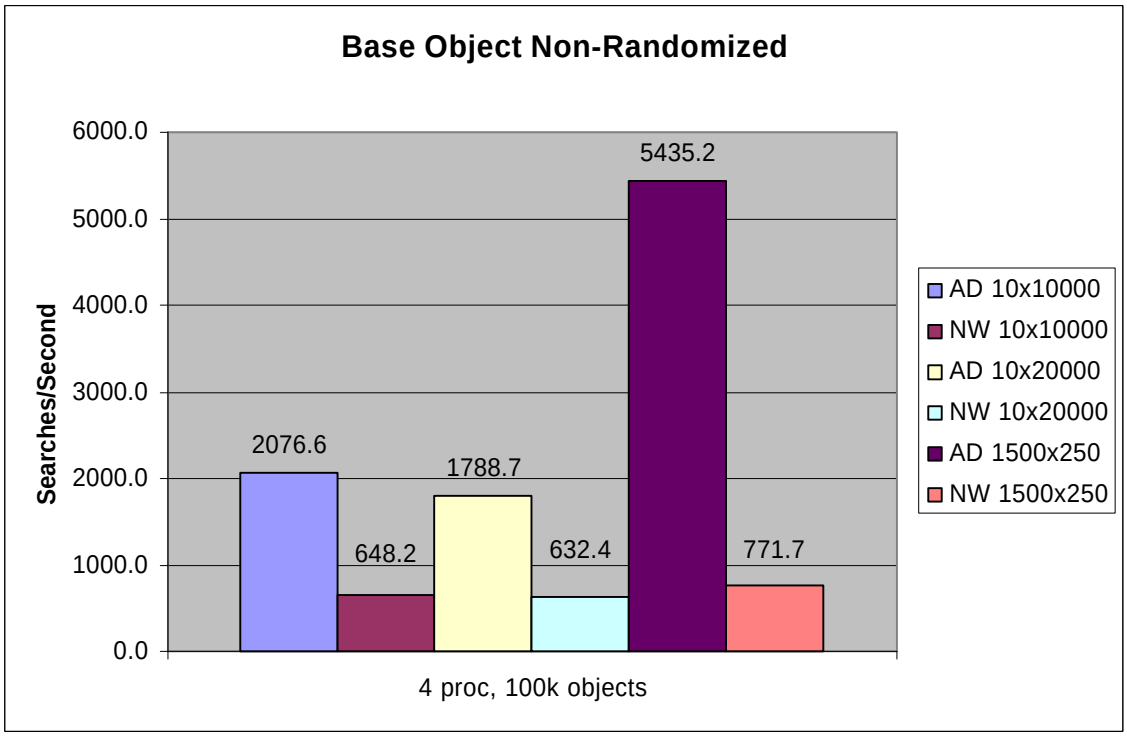
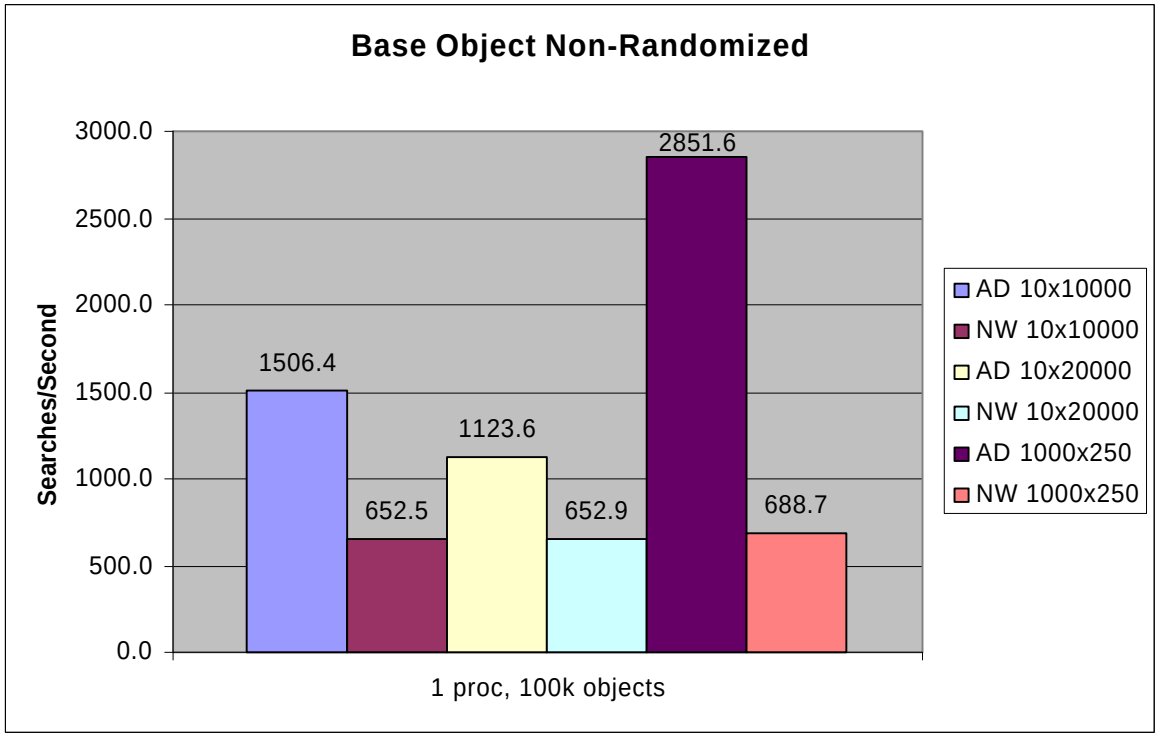


Figure 10: BS, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.2.2 Base Object Randomized

Figure 11: BSD, 4 Processors

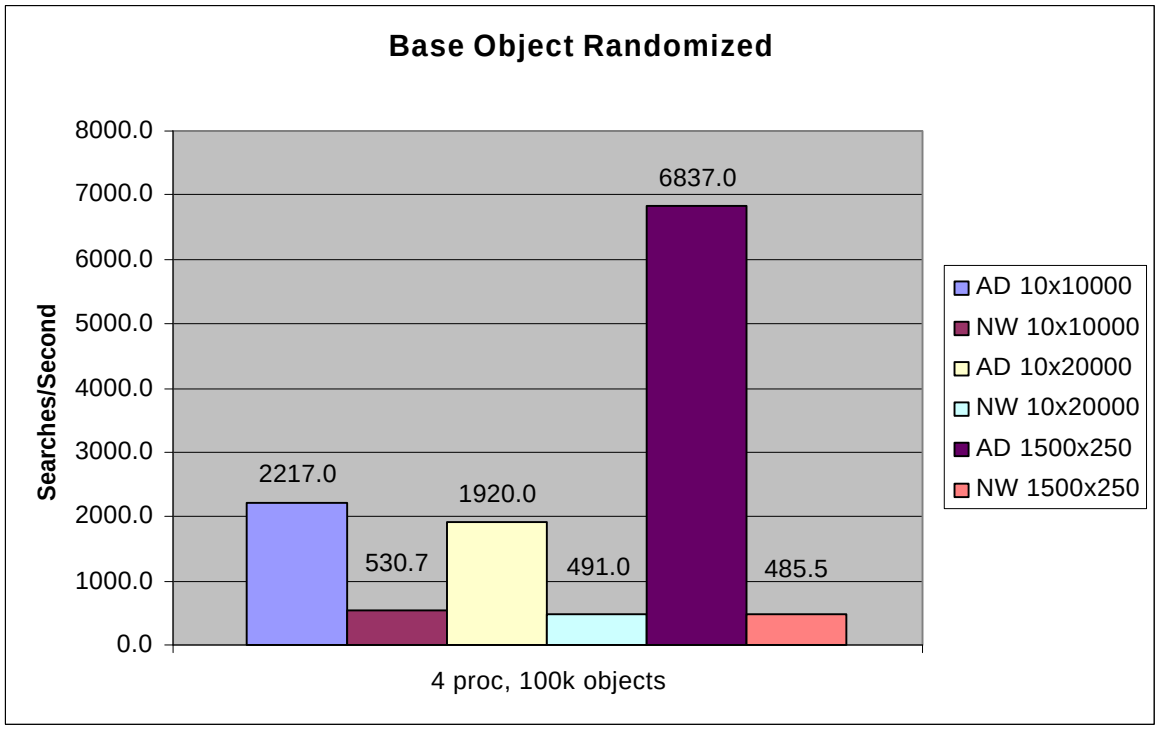
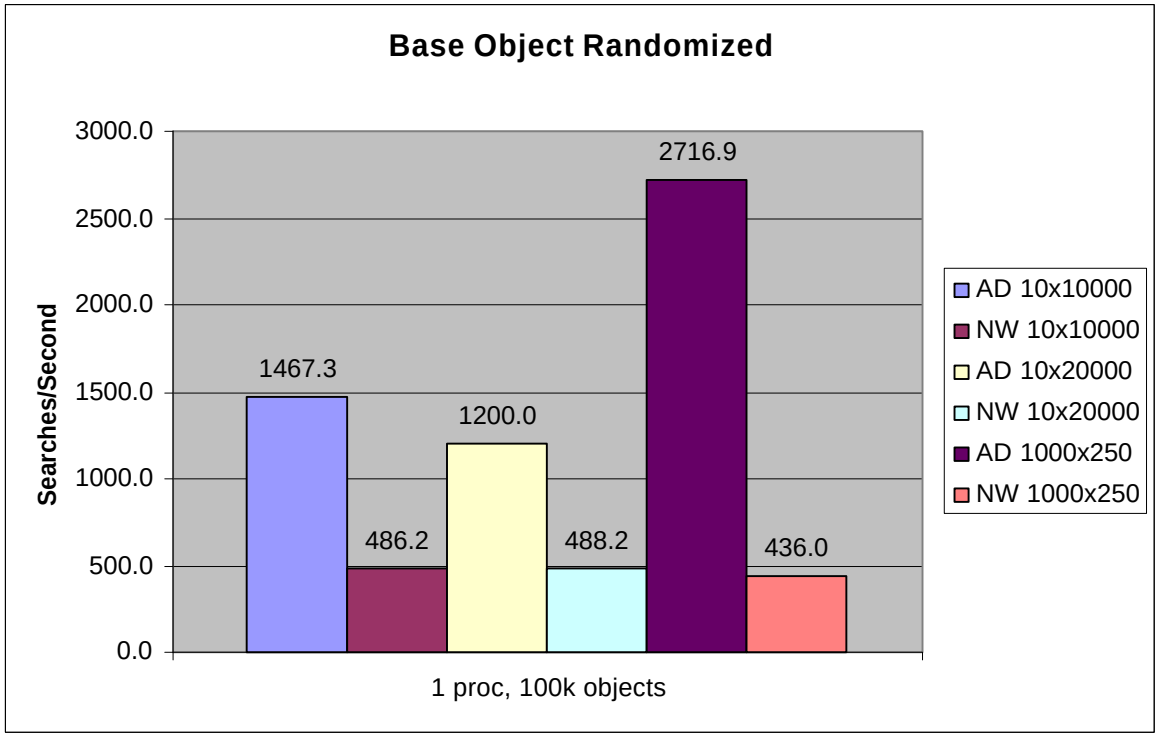


Figure 12: BSD, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.2.3 Sub-Tree Search Randomized

Figure 13: SSD, 4 Processors

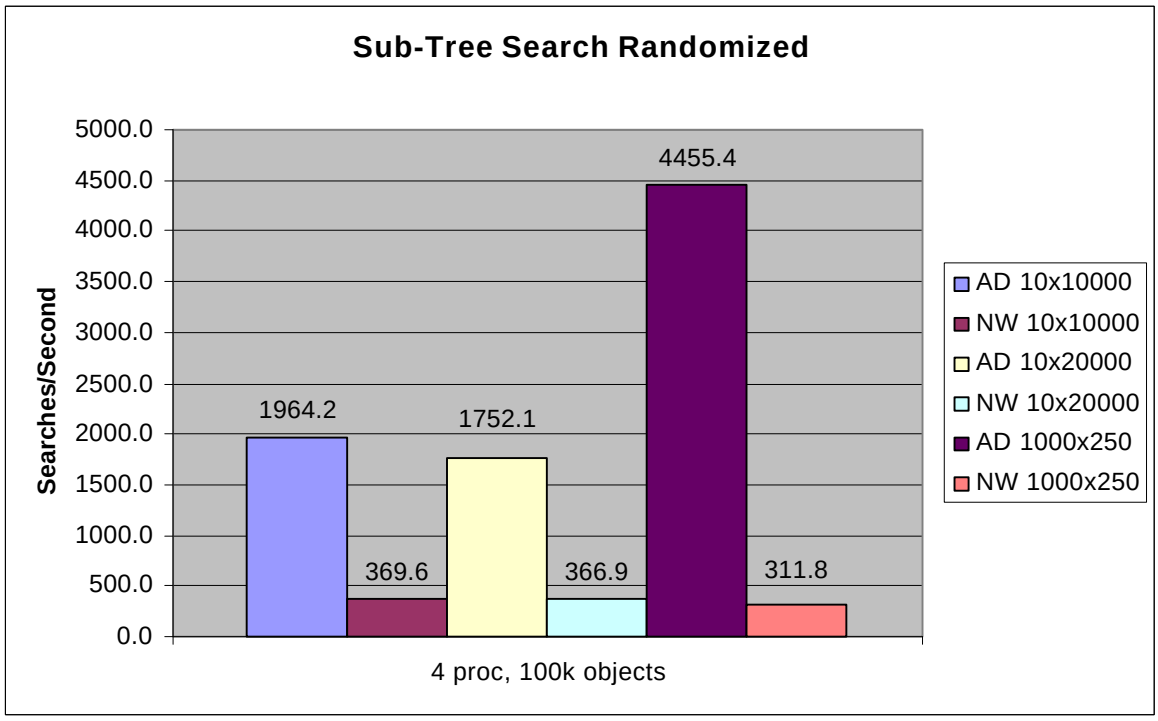
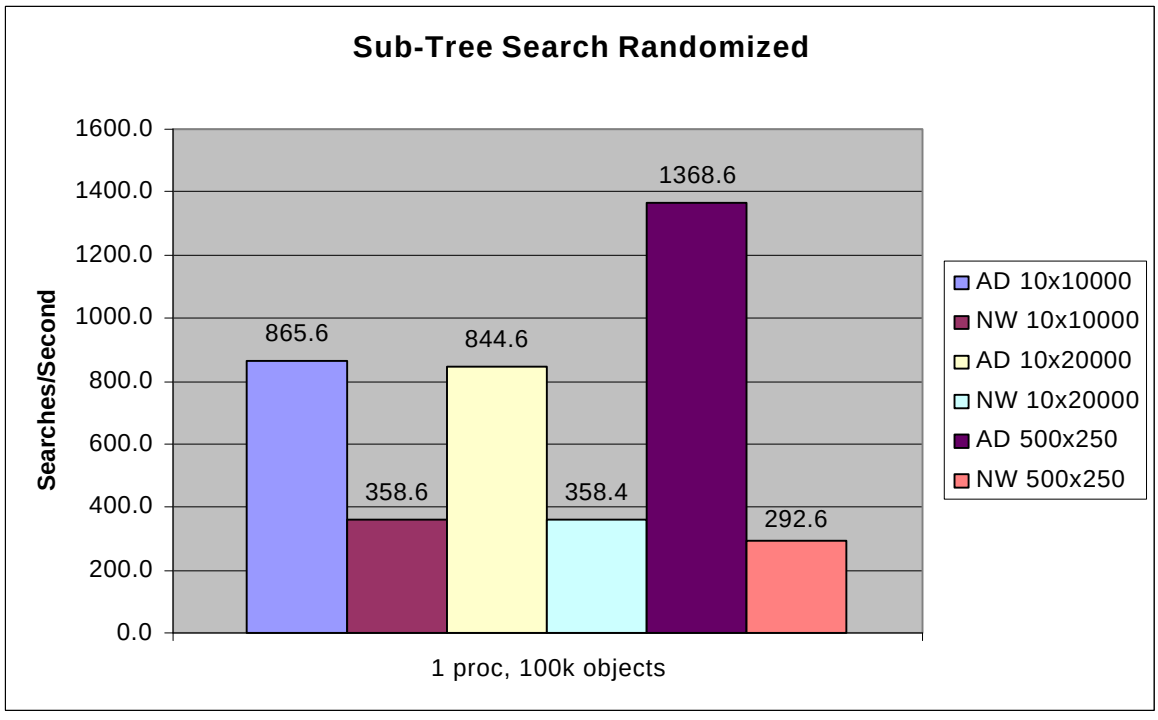


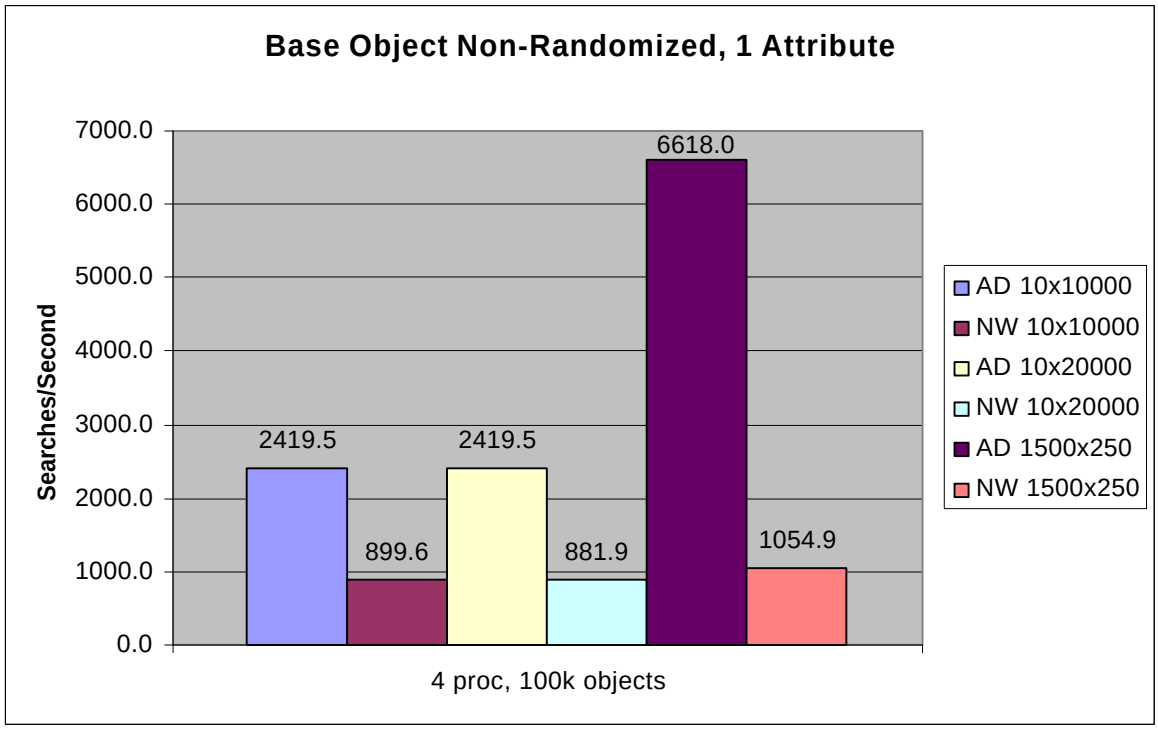
Figure 14: SSD, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

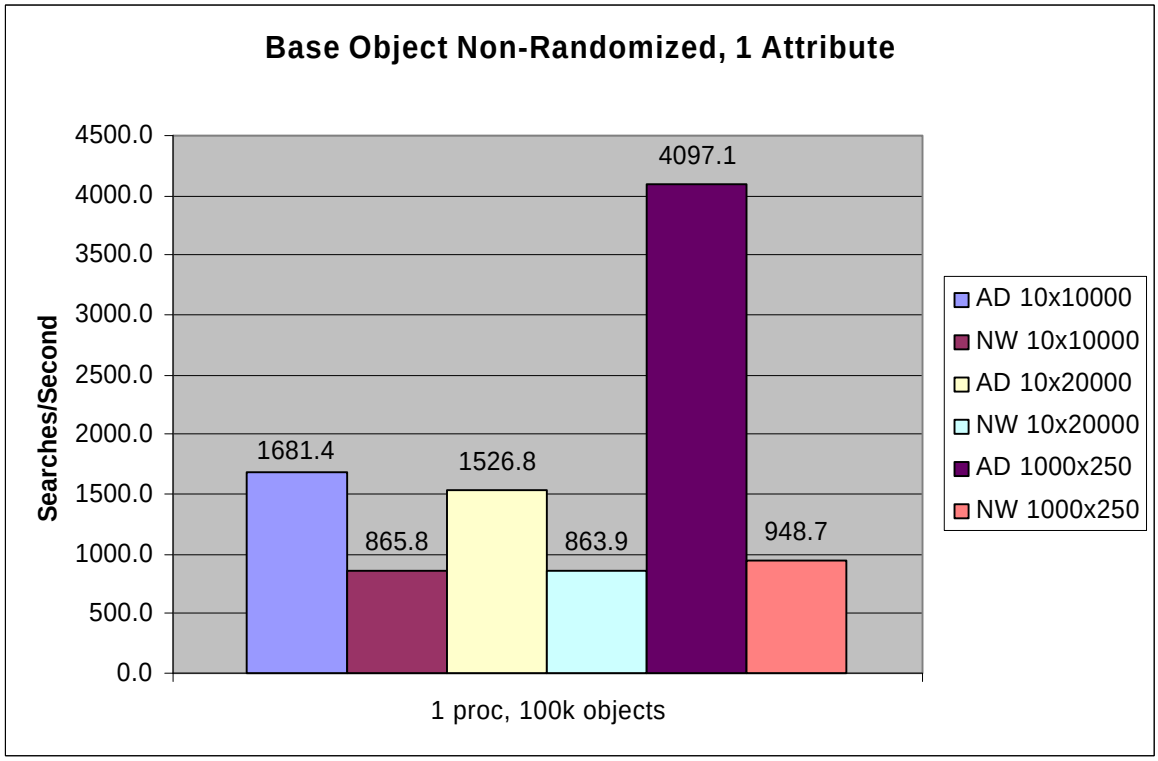
3.2.4 Base Object Non-Randomized, 1 Attribute

Figure 15: BS1A, 4 Processors



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Figure 16: BS1A, 1 Processor



At 1 million objects, the objects were created using custom PERL scripts. These objects were loaded using Bulkloader on NDS 8, and LDIFDE on Active Directory.

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.2.5 Base Object non-Randomized

Figure 17: BS, 4 Processors

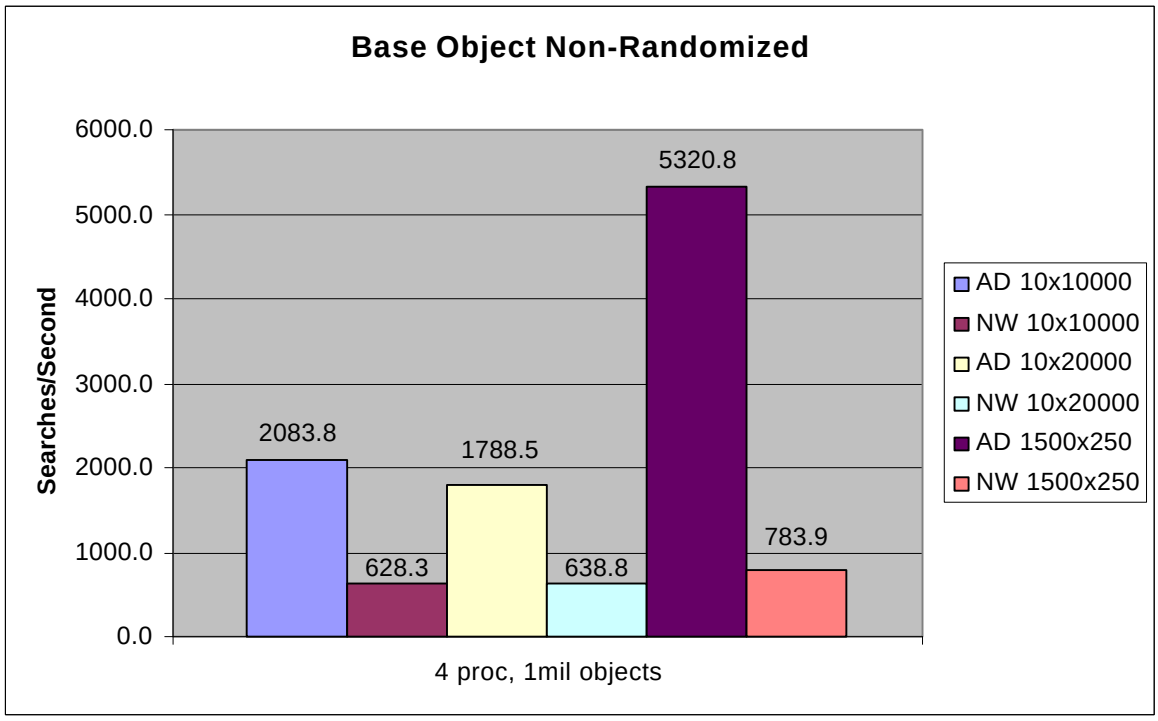
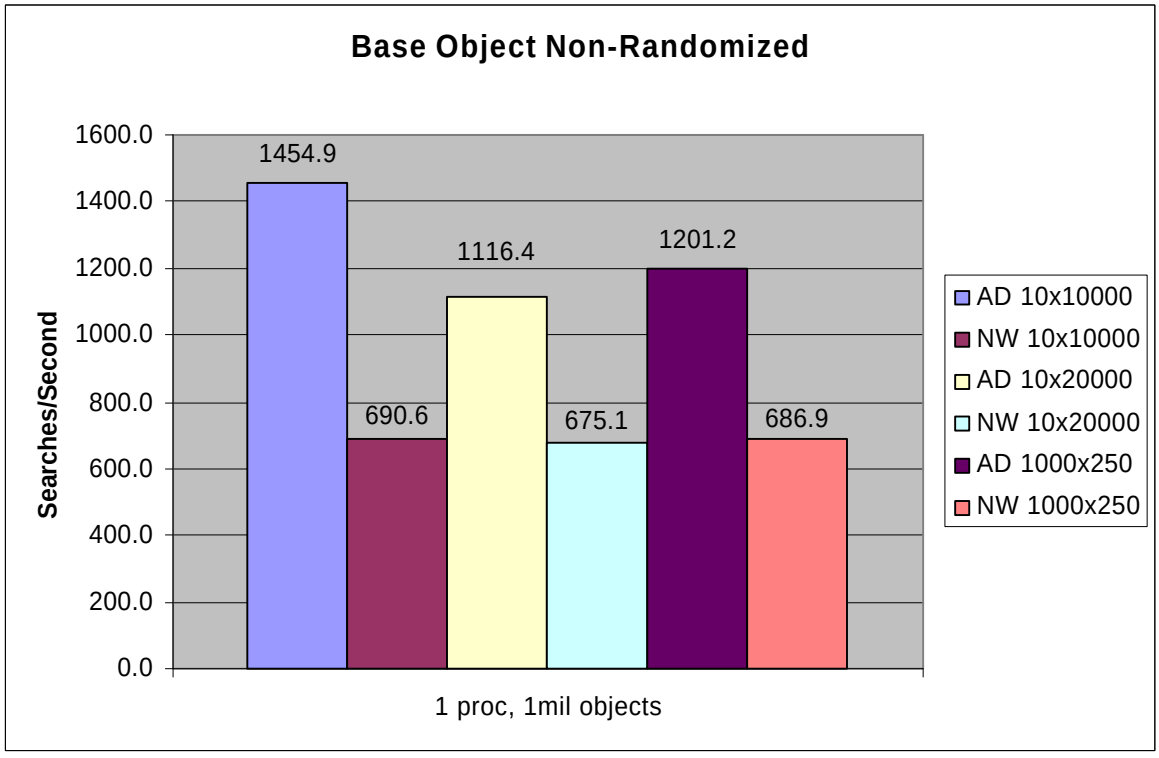


Figure 18: BS, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.2.6 Base Object Randomized

Figure 19: BSD, 4 Processors

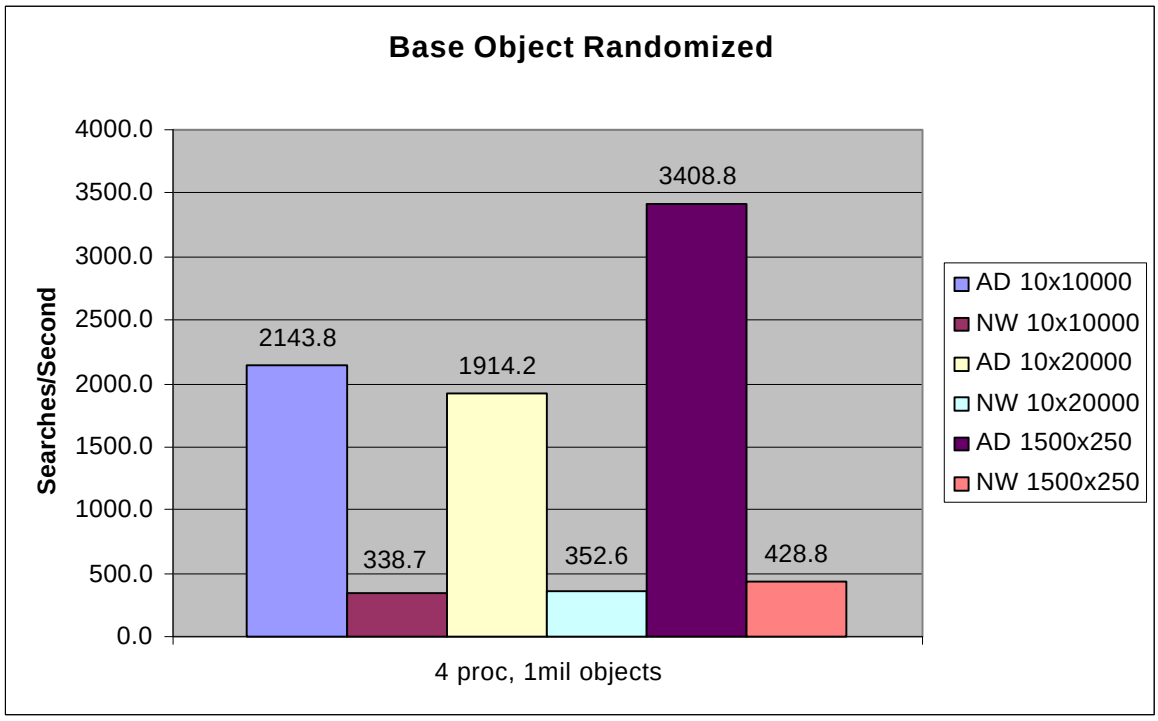
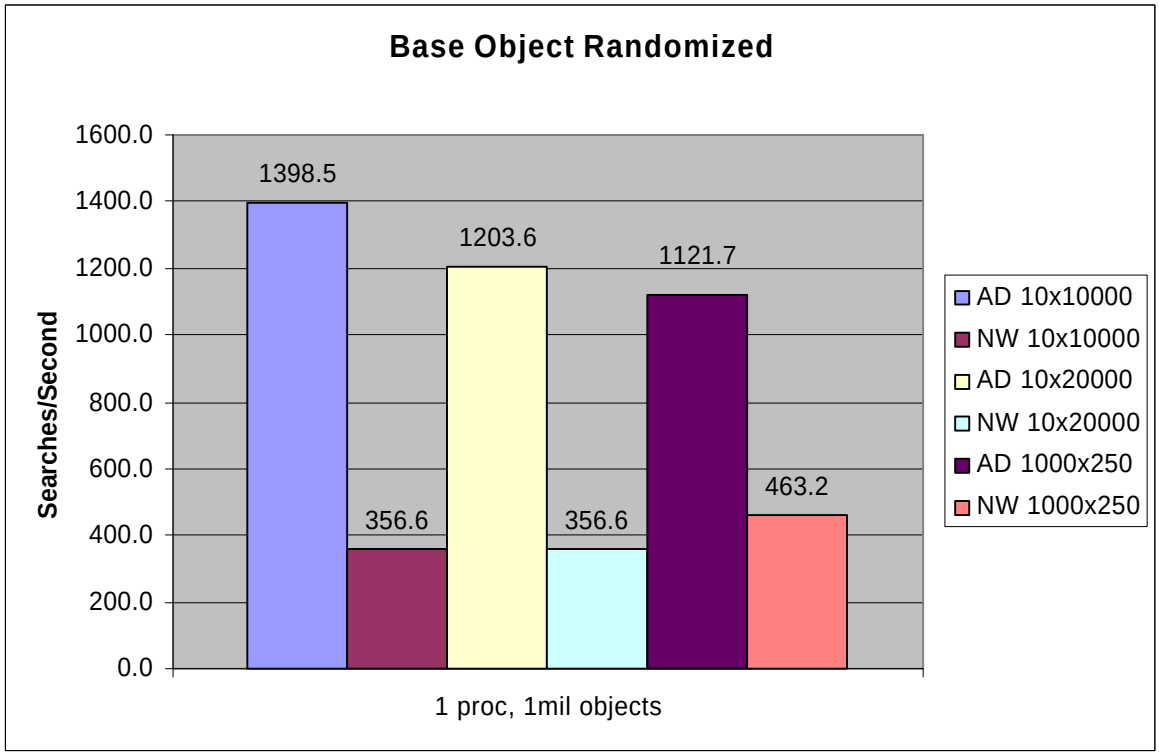


Figure 20: BDS, 1 processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.2.7 Sub-Tree Search Randomized

Figure 21: SSD, 4 Processors

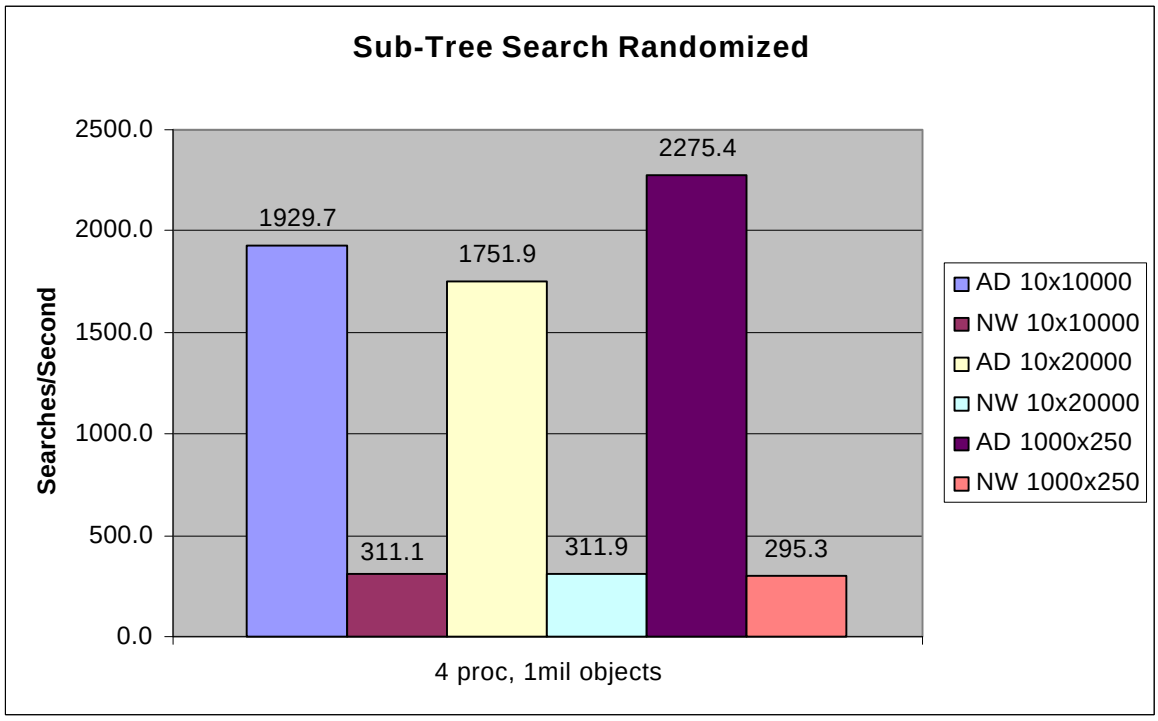
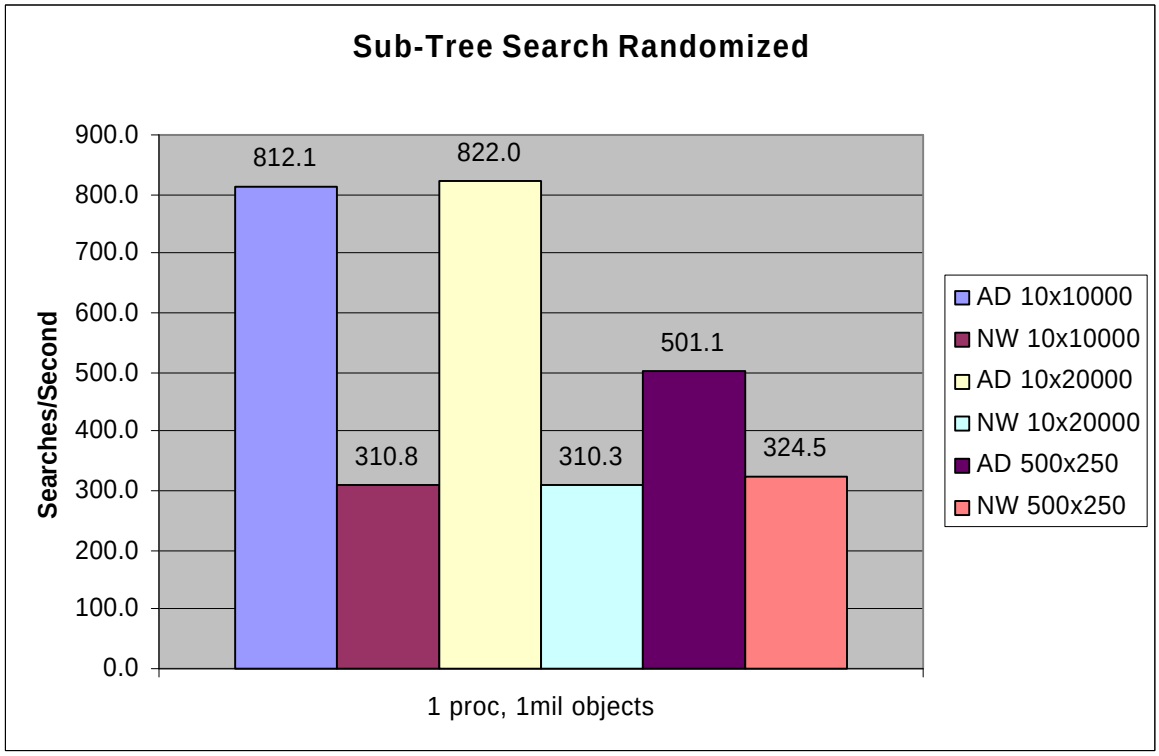


Figure 22: SSD, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.2.8 Base Object Non-Randomized, 1 Attribute

Figure 23: BS1A, 4 Processors

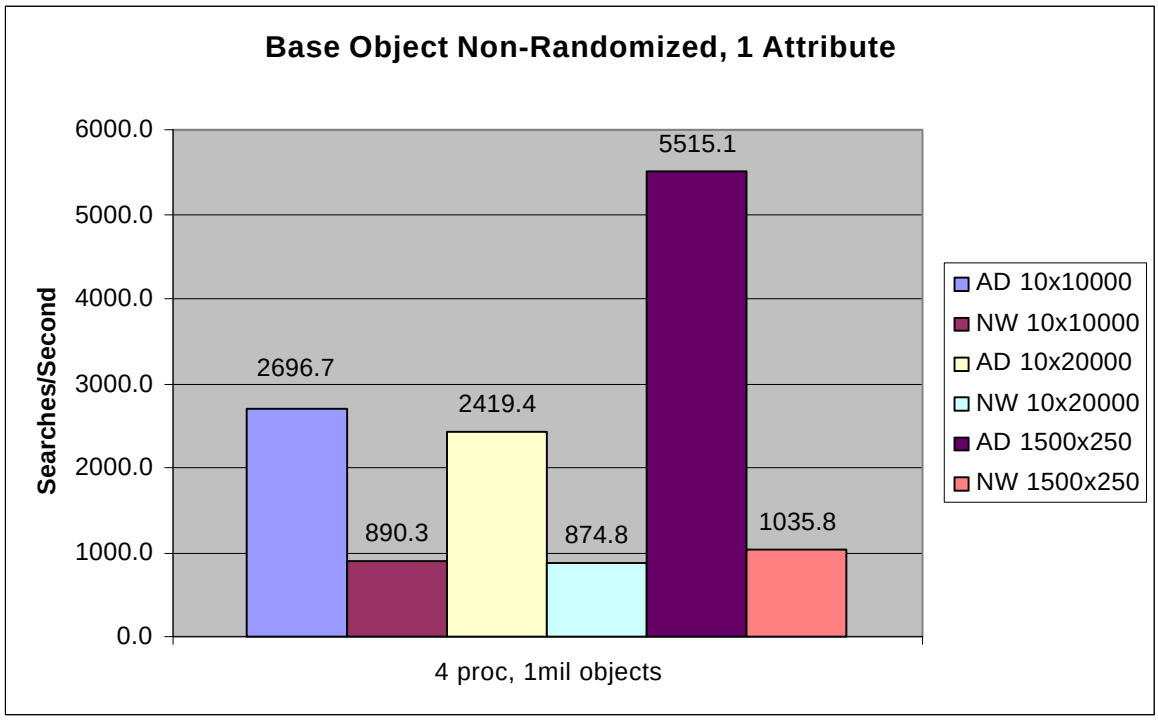
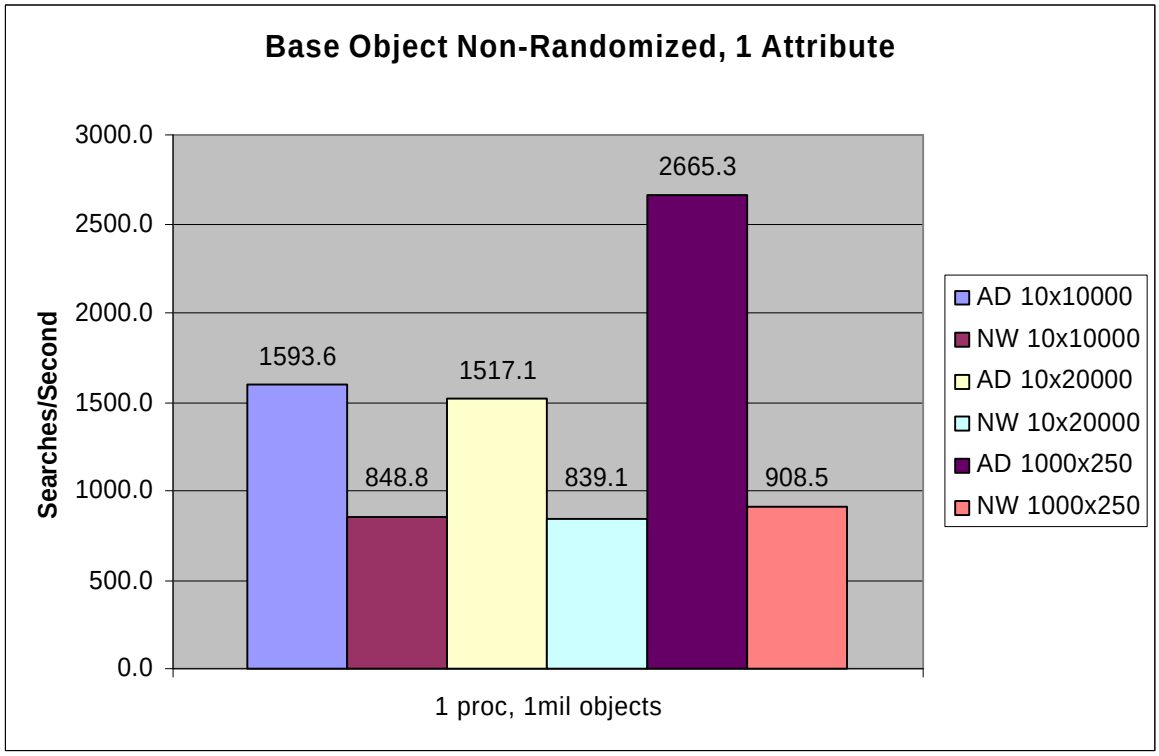


Figure 24: BS1A, 1 Processor



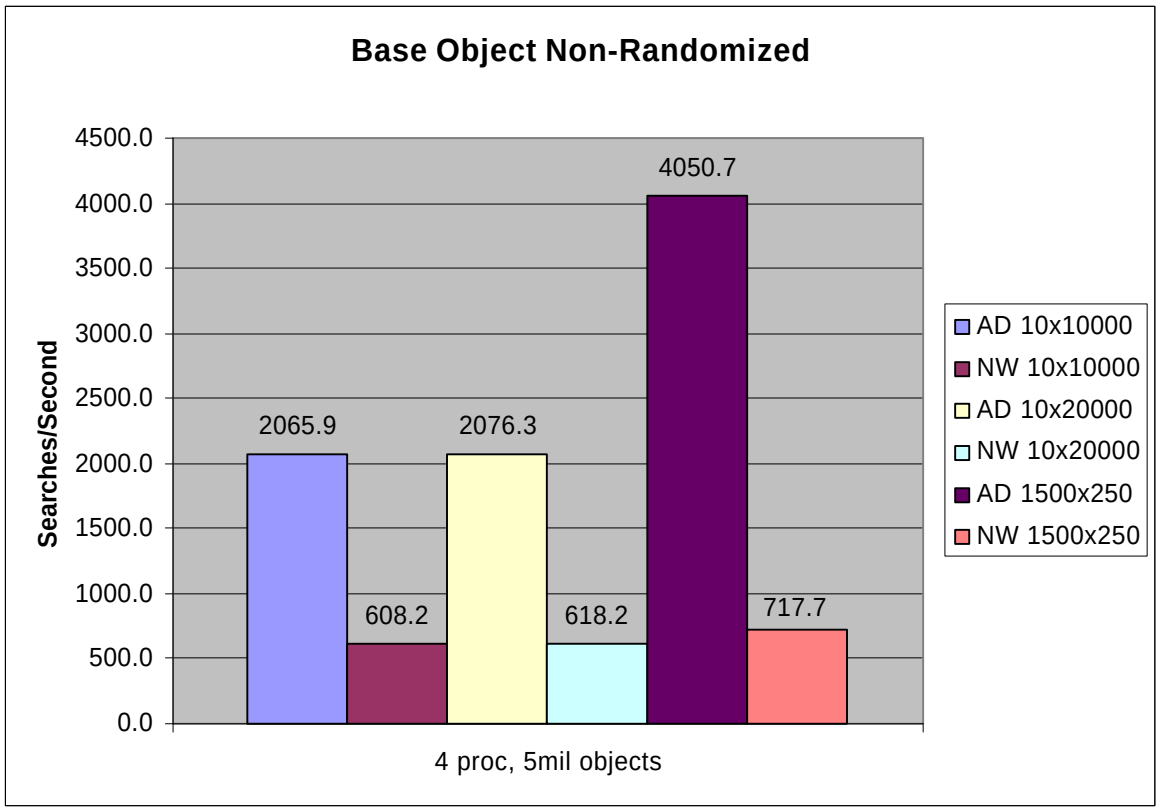
KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.3 5 Million Object Performance Testing

At 5 million objects, 3 million objects were created using custom PERL scripts. These objects were "seed" data and all 3 million were searchable. An additional 2 million objects were created and loaded in the directory. These objects are identical in type, but would not be searched. All objects were randomly dispersed throughout the OU's. All objects were loaded using Bulkloader on NDS 8, and LDIFDE on Active Directory.

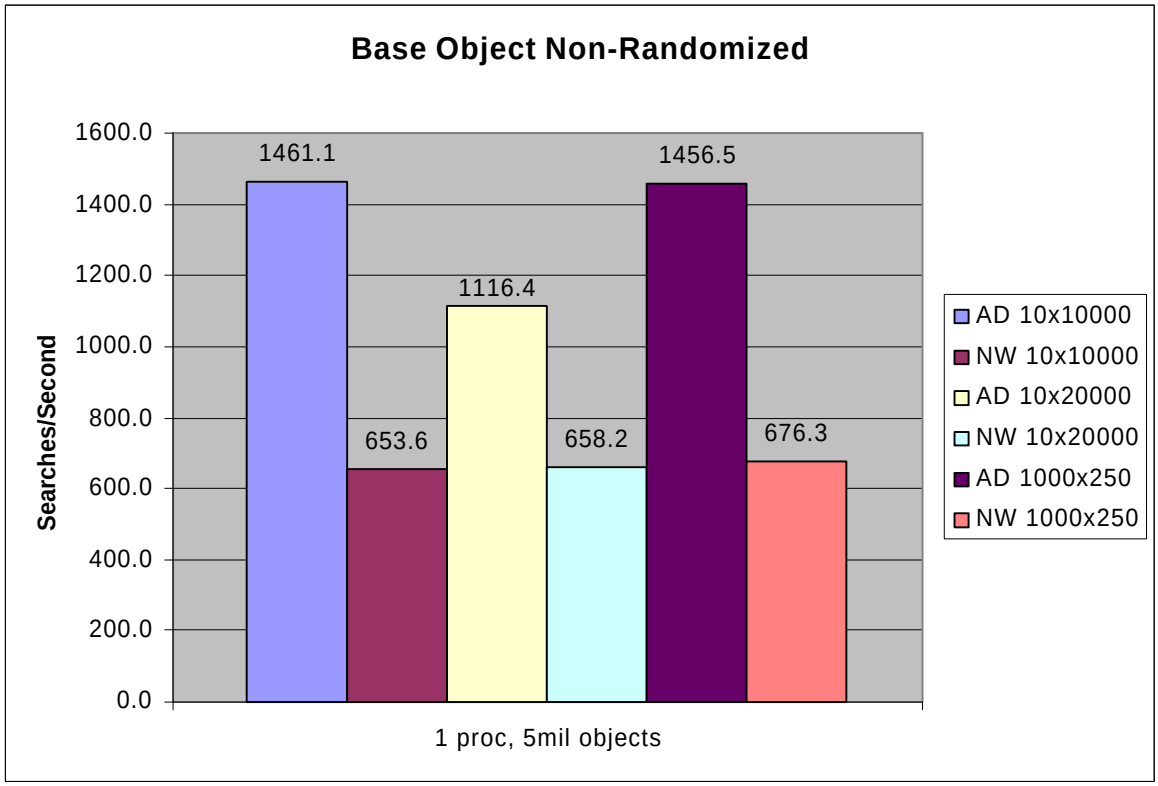
3.3.1 Base Object Non-Randomized

Figure 25: BS, 4 Processors



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

Figure 26: BS, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.3.2 Base Object Randomized

Figure 27: BSD, 4 Processor

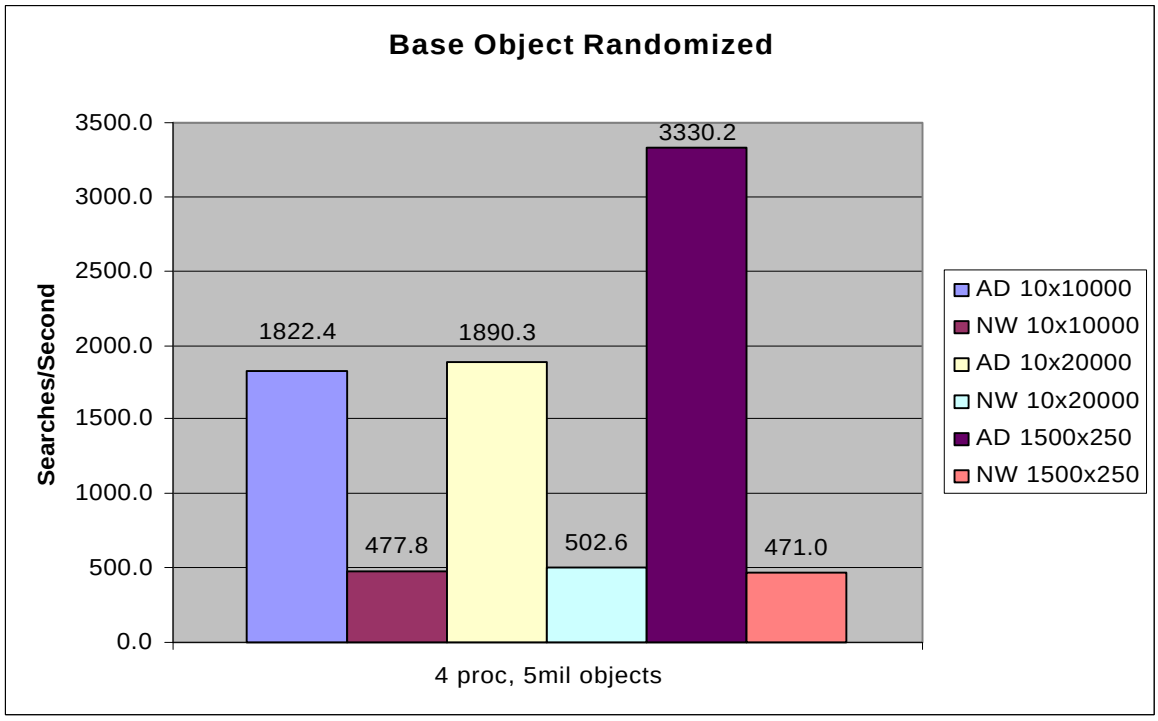
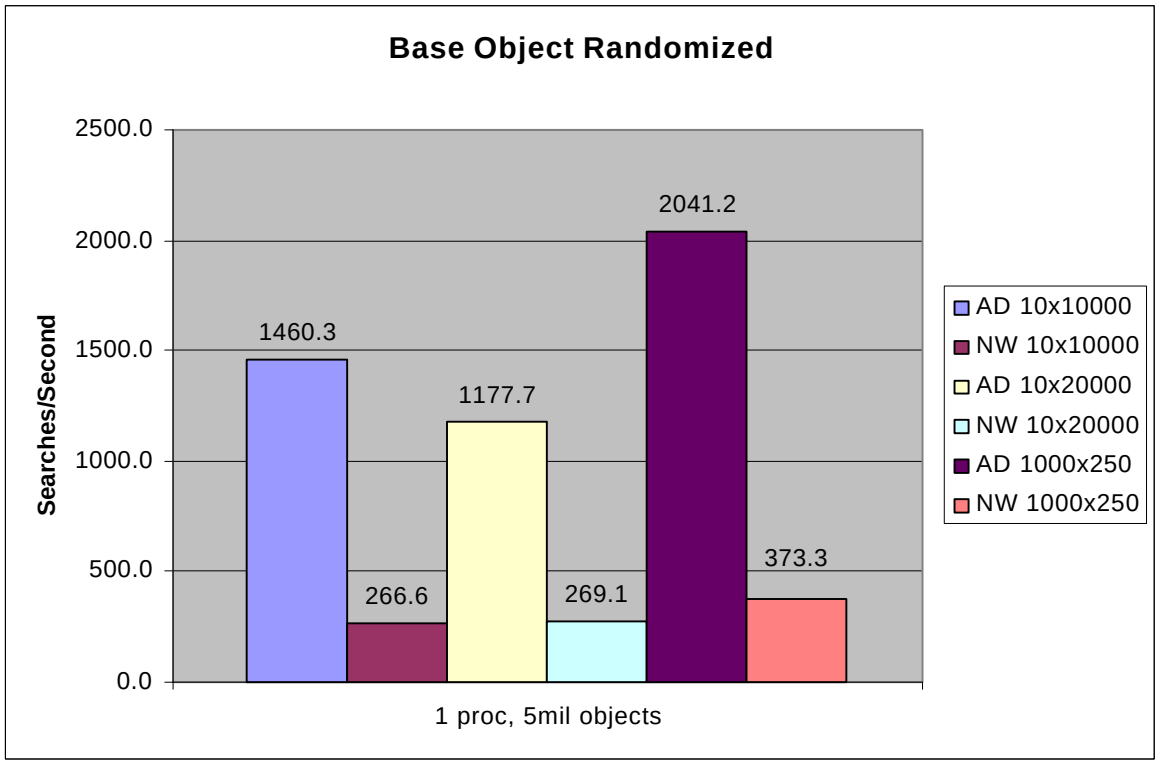


Figure 28: BSD, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.3.3 Sub-Tree Search Randomized

Figure 29: SSD, 4 Processors

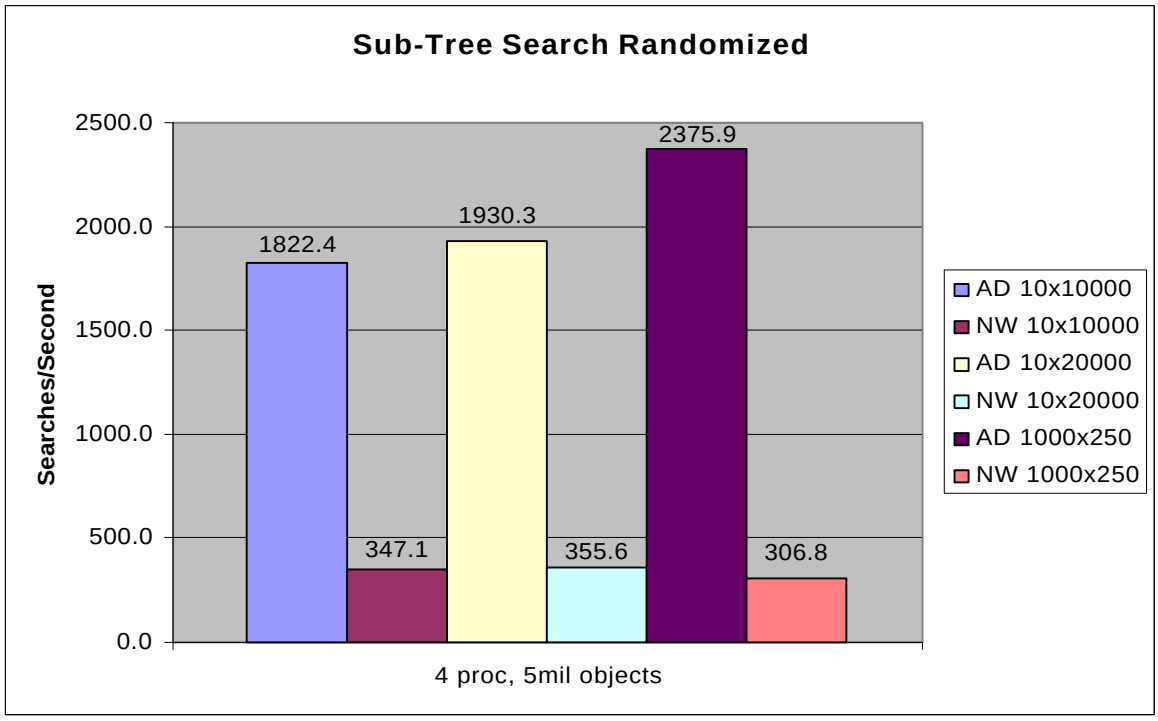
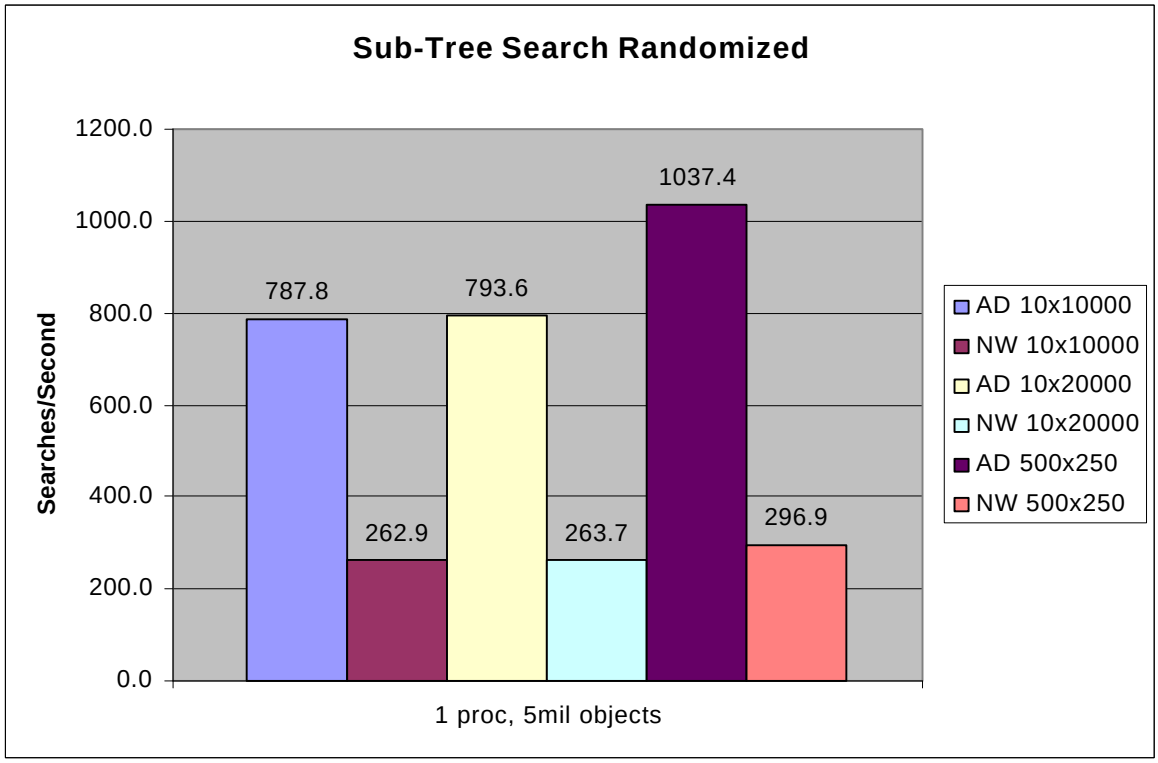


Figure 30: SSD, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.3.4 Base Object Non-Randomized, 1 Attribute

Figure 31: BS1A, 4 Processors

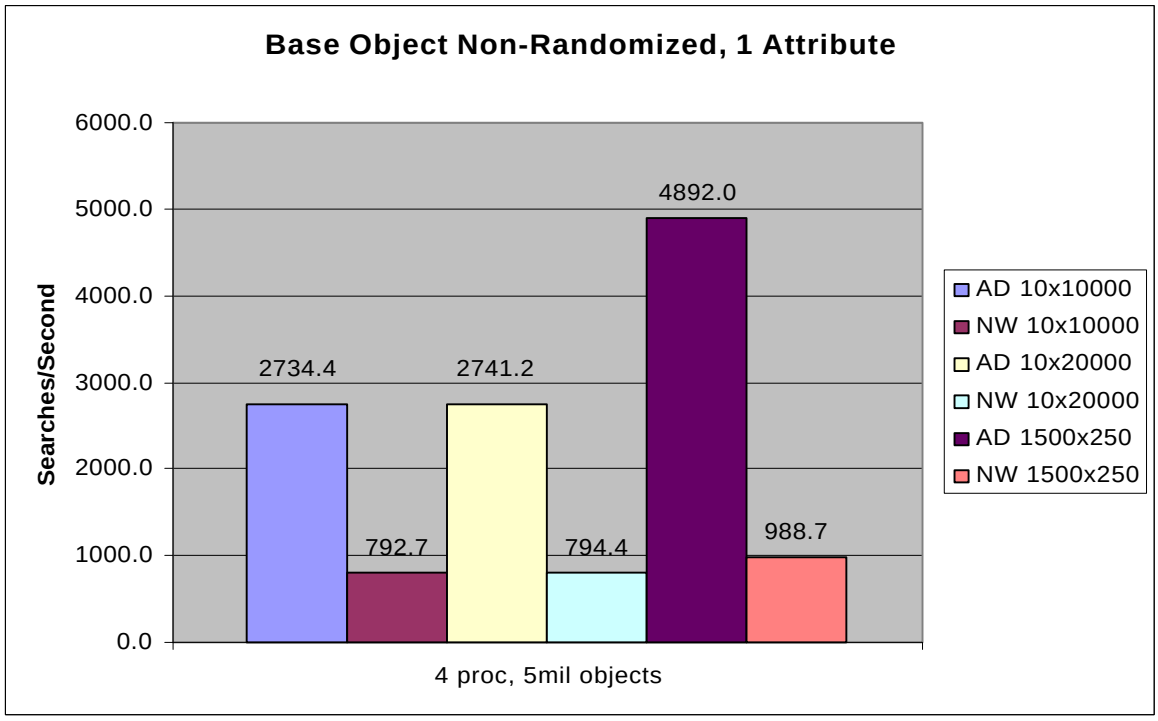
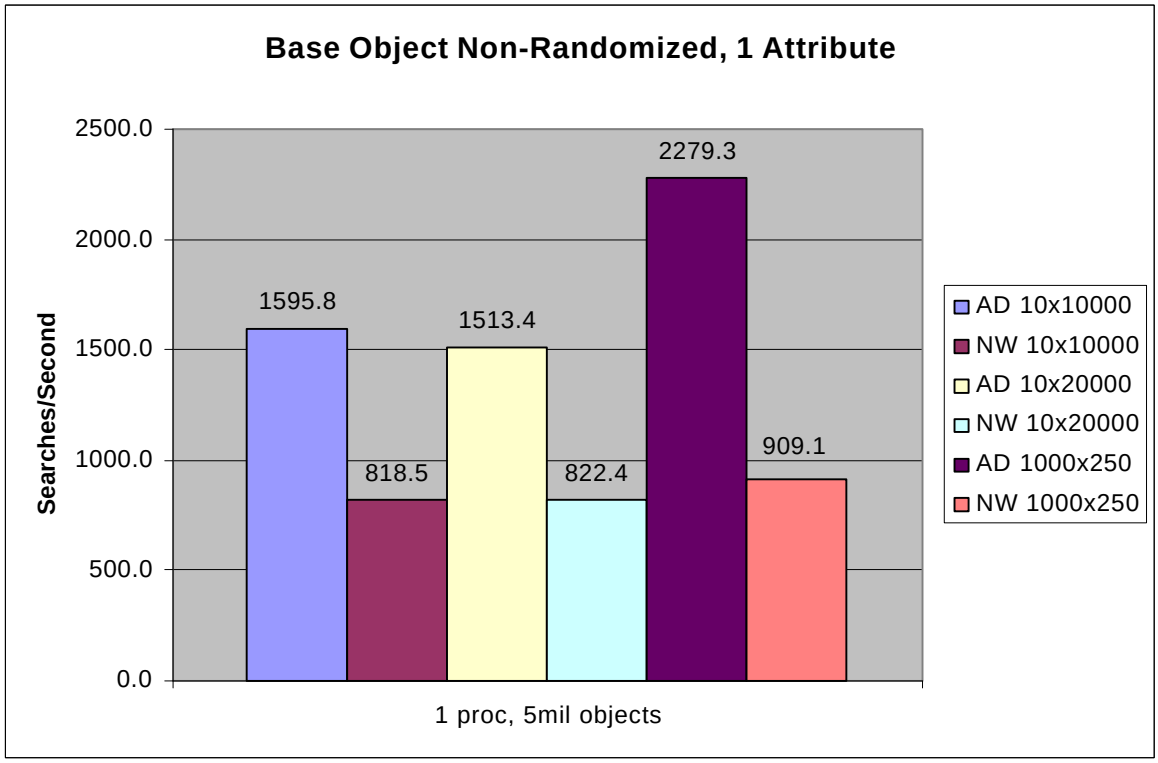


Figure 32: BS1A, 1 Processor



KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

3.4 Test Results Summary

Microsoft Active Directory on Windows 2000 Server was able to provide much higher LDAP search throughput than NDS 8 on NetWare 5.1 in all test scenarios, across all directory sizes tested, and on both single and four-processor configurations. As expected, the LDAP search performance of NDS 8 on NetWare 5.1 showed no improvement on the 4 processor system. Active Directory, on the other hand, exhibited significant performance gains when running on the SMP server. Both directories were able to maintain the ratio of its performance across the larger directory sizes.

The only caveat that Active Directory exhibited was its tendency to refuse additional connections if the CPU was near 98-100%. This posed a problem with the benchmark since it wasn't designed to handle these types of exceptions. Although this could pose a problem in an environment where the directory server was being over-utilized, most LDAP clients will retry a number of times after the connection is refused.

4.0 OPERATING SYSTEM TUNING

The testing was conducted on the following hardware configurations for both NetWare and Microsoft Active Directory. Performance scripts were run on the same SUN boxes for both configurations.

4.1 NetWare Installation

NetWare was installed from shipping source using default installation parameters. The following SET parameters where then modified:

set Maximum Pending TCP Connection Requests = 4096
set Maximum Packet Receive Buffers = 10000
set Minimum Packet Receive Buffers = 3000
set Maximum Physical Receive Packet Size = 2048
set Maximum Concurrent Disk Cache Writes = 2000
set Dirty Directory Cache Delay Time = 0
set Maximum Concurrent Directory Cache Writes = 500
set Maximum Directory Cache Buffers = 200000
set Maximum Number of Internal Directory handles = 1000
set Maximum Number of Directory Handles = 100
set Maximum Record Locks Per Connection = 10000
set Maximum Record Locks = 100000
set Maximum Outstanding NCP Searches = 500
set enable file compression = off
set immediate purge of deleted files = on

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

4.2 Windows 2000

Windows 2000 was installed from Build 2194 source using default installation parameters. Active Directory was then installed using default parameters. When testing Windows 2000 with multiple processors, the multi-proc kernel was utilized, and the server had only four processors installed. When testing on the single processor configuration, all processors except one were removed, and the uni-proc kernel and associated DLL's were installed.

The changes to default installation procedures were:

Windows 2000
Indexing – Not installed.
IIS – Not Installed.
Script Debugging – Not Installed
Network Monitoring Tools – Installed
Netbios over IP – Disabled
LMHosts – Disabled
The drives containing the Active Directory files and log files were set to 8k allocation.

Active Directory
When installing, permissions were set to WIN2000 servers only.

5.0 IMPORTANT INFORMATION

KeyLabs certifies that the comparison tests described in this report were performed at its corporate testing facility in Lindon, Utah. The results reported herein represent the actual results of the testing.

To the best of KeyLabs' knowledge, these results are accurate and reproducible by any party who deploys the same configuration specified for this comparison and executes the same tests. Notwithstanding, KeyLabs reserves the right to modify this report without notice to correct errors or omissions.

Microsoft Corporation contracted KeyLabs, Inc. to perform this comparison test and publish the results.

5.1 KeyLabs Certification

Any access to or use of this Report is conditioned on the following:

1) The information in this Report is subject to change by KeyLabs without notice.

2) The information in this Report is believed by KeyLabs to be accurate and reliable, but is not guaranteed. All use of and reliance on this Report are at your sole risk. KeyLabs is not liable or responsible for any damages, losses or expenses arising from any error or omission in this Report.

3) NO WARRANTIES, EXPRESS OR IMPLIED ARE GIVEN BY KEYLABS. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT ARE DISCLAIMED AND EXCLUDED BY KEYLABS. IN NO EVENT SHALL KEYLABS BE LIABLE FOR ANY CONSEQUENTIAL,

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205

INCIDENTAL OR INDIRECT DAMAGES, OR FOR ANY LOSS OF PROFIT, REVENUE, DATA, COMPUTER PROGRAMS, OR OTHER ASSETS, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

4) This Report does not constitute an endorsement, recommendation or guarantee of any of the products (hardware or software) tested or the hardware and software used in testing the products. The testing does not guarantee that there are no errors or defects in the products, or that the products will meet your expectations, requirements, needs or specifications, or that they will operate without interruption.

5) This Report does not imply any endorsement, sponsorship, affiliation or verification by or with any companies mentioned in this report.

All trademarks, service marks, and trade names used in this Report are the trademarks, service marks, and trade names of their respective owners, and no endorsement of, sponsorship of, affiliation with, or involvement in, any of the testing, this Report or KeyLabs is implied, nor should it be inferred.

KeyLabs www.keylabs.com 385 South 520 West Lindon, UT 84042 (801)226-8200 fax:(801)226-8205