White Paper

Agiloft

Agiloft Scalability and Redundancy

Table of Contents

Introduction

This paper reviews Agiloft scalability from the perspective of actual use on a hosted server and benchmarks on standard hardware. This analysis is followed by a discussion of the load that is placed on the system by typical users and a brief summary.

The benchmark section is focused on operations that are common in production use, such as logging into the system, creating or updating a record and searching for an existing record, and analyzes how scalability increases with the number of CPU cores.

It does not address the time taken to execute infrequent administrative actions such as creating a custom table, since these are intrinsically single threaded and are almost unaffected by the number of processors.

Performance on Hosted Server

Figure 1: Real World Performance

	Avg Load 9am – 5pm	Peak Load 9am – 5pm
Hosted Server with 1512 users	0.68	1.17

Comments:

These numbers were obtained from a server configured with dual Xeon X5355 processors, RAID 10 SATA hard drives and 24G RAM with 41 hosted knowledgebases (KB) and an average of ~36 users per KB for a total of 1483 users. The 41 KB's provide a broad cross-section of use patterns and the load remains low throughout the business day.

Performance typically starts to degrade once the load reaches a number between 2.5 and 3.5 so this server could support a significant number of additional users.

All customers on this particular server are located in North America, so the load is highest between 5am and 5pm PST. A machine that served an even distribution of international customers could support about twice as many users without increasing the peak load.

Benchmarks

System configuration used for benchmarks

Database: MySQL

Hardware: Dual Xeon E5440 processors, 24 G RAM, 3 Seagate SAS drives

OS: Suse Linux 11

Agiloft Release: 2008_03-12507-127-r82842

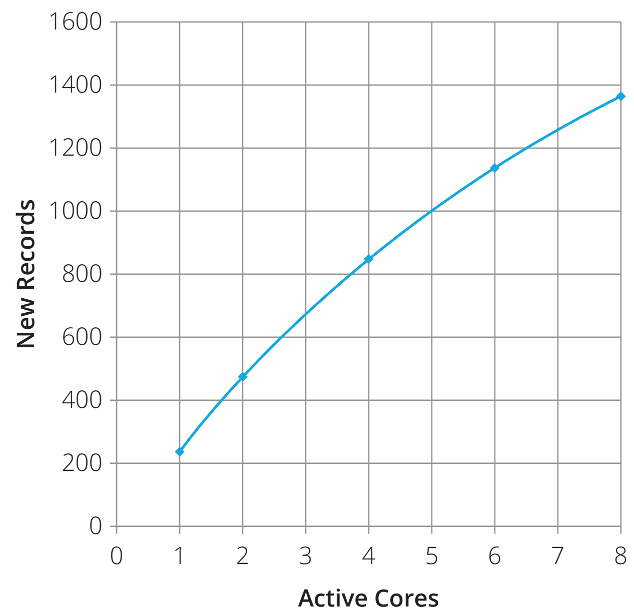
Comments:

The Nehalem server processors, released in Q1 2009, are expected to increase performance by 40-80%.

SAS hard drives provide ~15% better raw server performance for most operations, but except on high volume systems, this is not noticeable for users accessing the system over the Internet because server-side performance is dwarfed by the 1-2 second Internet delay for each user transaction.

Agiloft also supports MS SQL and it provides the same performance as MySQL to within 15%. On average MySQL is slightly faster than MS SQL for most tests/loading conditions.

Figure 2a: New records per minute on dual E5440 processors



Comments:

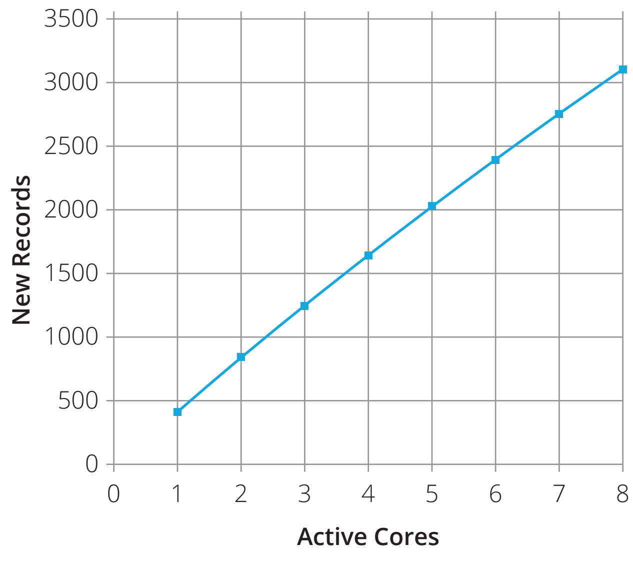
This test was run on the default Demo KB. Performance will increase or decrease depending upon the size of the record, the complexity of the KB structure and the number of active business rules.

The time required to edit an existing record is typically the same as the time required to create a new one.

Customers can run the same test on their hardware by selecting Setup/Performance Test from the admin console.

As indicated by figure 3a, the number of records in the database does not have a measurable effect on performance, so a system with dual E5440 processors (8 cores) can generate over 80,000 records per hour until it runs out of disk space.

Figure 2b: New records per minute on dual E5530 processors



Comments:

This test illustrates the performance gains that are available from an alternate hardware configuration of dual i7 processors and dual Intel X25-E SSD hard drives.

With hyper-threading turned on, the number of records per minute jumped to 3870.As indicated by figure 3a, the number of records in the database does not have a measurable effect on performance, so a system with dual E5530 processors (8 cores) system can generate over 232,000 records per hour until it runs out of disk space.

Figure 3a: Effect of the number of records on database search performance

Number of records in database	Time to find text string in database
10,000	< 1 second
1,000,000	< 1 second
10,000,000	< 1 second

Comments:

These numbers reflect the fact that Agiloft incorporates a database-independent full text search engine.

There are operations available only to system administrators, such as performing a mass update of all records in the knowledgebase, that will vary linearly with the number of records in the database.

Figure 3b: Effect of the number of records on file search performance

Number of records in database	Time to find text string in attached file
10,000	< 1 second
1,000,000	< 1 second
10,000,000	< 1 second

Comments:

These numbers reflect the fact that Agiloft incorporates a database-independent full text search engine that indexes attached files.

Figure 4a: Number of logins per minute to the End User Interface

Number of active processor cores	Number of end user logins per minute
1	375
2	700
4	1150
6	1469
8	1750

Comments:

These numbers reflect the number of unique users that may login to the Agiloft end user interface per minute. Comparing with figure 2, we see that when a user logs into the system, it imposes a load comparable to creating a single record. As detailed in figure 5, any number of users may be logged into Agiloft and unless they are actively doing something, they do not impose a measurable load on the system.

Figure 4b: New logins per minute to the Power User interface on E5440 processors

Number of active processor cores	Number of power user logins per minute
1	153
2	284
4	466
6	597
8	712

Comments:

These numbers reflect the number of unique users that may login to the Agiloft power user interface per minute. As detailed in figure 5, any number of users may be logged into Agiloft and unless they are actively doing something, they do not impose a measurable load on the system. The number of new logins per minute on dual E5530 processors exceeds 1,500.

Figure 5: Effect of the number of logged in users

Number of logged in users	Time to open a new record
100	< 1 second
1,000	< 1 second
10,000	< 1 second
100,000	< 1 second

Comments:

These numbers were obtained with a browser on the same local network as the server with one active user. When connecting over the Internet a delay of 1 to 2 seconds for network connectivity is typical.

While the presence of a logged-in user does not affect performance, users who are actively creating or editing records will affect performance once the number of requests approaches the rate per CPU shown in figure 2. If 1,000 users were creating new records at the rate of one per user per hour, there would not be a significant performance impact. 1,000 users who were each attempting to create new records at the rate of one per user per minute would bring an 8-core system to a crawl.

Although the number of logged in users does not directly affect performance unless they are active, it does affect the amount of RAM required by JBoss. Each logged in user uses about 5K of RAM, so a server that is intended to support 100,000 logged-in users would be using 5G RAM to hold the login information in memory. Naturally, this requirement only applies to the number of logged-in users. Users who exist in the database but are not logged in do not use any RAM.

System Requirements

Agiloft is built on EJB architecture for scalability and has been tuned through years of production use.

Provided the system is configured with at least 8G of RAM, very little time is spent on I/O and performance depends primarily upon the CPU. Because the system is EJB based, integer performance and the amount of cache in the CPU have a strong impact on performance and recent generations of the Intel chipset are recommended over Opteron or Sparc processors.

Load Balancing

Load balancing can be applied at the database level, the application server level and by splitting the EJB and database servers across different machines.

Separate database and EJB servers

About 55% of CPU resources are spent on Java processes and 45% on database processes. By placing the application server and database on separate machines, it is therefore possible to significantly increase scalability.

However, it is important to distinguish between scalability for peak loads and performance under typical loads. Splitting the database and application server across different machines will increase scalability but it also increases the communication delay between the database and application server, so while this method will improve performance under peak-load conditions when all cores are active concurrently, performance under typical load conditions will decrease slightly.

Multiple database servers

The database can be configured to distribute reads among multiple systems, but of course writes must be applied to all systems concurrently. As in the example above, this can increase scalability under peak loads, particularly if the system has an unusually heavy load from SQL read statements, but the introduction of the load balancer overhead will decrease performance under typical loads.

An alternative to deploying multiple database servers is to configure the machine with SSD hard drives. These provide read times several times faster than traditional hard drives and seek times an order of magnitude faster, at a price point that is significantly lower than an additional server. Such a configuration not only improves scalability under peak loads, but also improves performance under typical loads.

Multiple application servers

The EJB architecture can be leveraged to cluster the application server across multiple machines and JSP server instances can also be distributed across multiple machines. However, these both carry a high communications overhead since the cache of each application server or JSP server instance must be synchronized. This is only practicable if the application servers are closely coupled by at least a 1Gps connection and in practice; any performance gain from CPU utilization is swamped by the connection overhead under typical load conditions. If the servers were connected over a WAN, performance would suffer severely. For the above reasons, this method of load balancing is no longer supported.

Hyper-Threading Performance

For customers running on the Intel i7 architecture, the re-introduction of hyper-threading provides a compelling alternative to load balancing for distributing the CPU load. Whereas prior generations of hyper-threading did little to improve performance, the increased cache sizes and improved architecture of the i7 processors have radically changed the situation.

For example, by turning on hyper-threading on a single i7 920 CPU, the number of new records that may be created per minute jumps from 833 to 1,200. A single E5530 processor, with hyper-threading turned on and an SSD hard drive, was benchmarked at 2,134 records per minute. Dual E5530 processors, with SSD hard drives were benchmarked at 3,870 records per minute.

Server Redundancy

For high availability configurations, DRDB provides full redundancy by mirroring the hard drives across multiple machines while heartbeat activates the necessary services on the secondary server if the primary dies. The IP address of the primary server is transferred to the secondary so that it transparently replaces the primary machine and the total service interruption time in the event of catastrophic hardware failure is less than 3 minutes. We have used this configuration with our hosted service for several years and provide pre-configured systems or instructions for configuring your own Linux hardware at no additional charge.

This methodology can be used if the secondary machine is located at a different facility; however DRDB proxy should be used in this case.

Infrastructure Redundancy

In addition to the application server redundancy detailed above, the infrastructure at the co-location facility is configured for high-availability as detailed below:

The Ethernet drop to the central switch is redundant and configured for hot-swap failover in the event that the co-location facility's router should fail.

The server blades are redundant and are connected via ISCSI to a hot-swap redundant file servers.

The file servers are configured with redundant power supplies and RAID 10 hard drives with hot-swap spares.

The blade server and file servers use EEC RAM.

The firewalls are redundant and configured for hot-swap failover.

The central switch is redundant and configured for hot-swap failover.

All components are subject to active monitoring and connected to remote power supplies.

Performance Analysis

System resources are only used when the user actually does something, such as creating or editing a record. As is typical for HTML based web products, the "overhead" from passive users is almost zero. For example, when users are filling out a web form, they are entirely passive from the perspective of the system and only become active on clicking the Finish button. This is important because it means that performance is not based on "how many users" there are, but "how many records those users are creating/editing per minute".

The average member of a sales team or support staff works on 3-8 records per day. If we assume that all users are at the top end of this range and work on 8 records per day, we would expect a sales/support staff with 1,000 full time agents to generate or edit 8,000 records per day, significantly less than our entry level system can handle in half an hour. Of course, it is necessary to add a safety factor for sudden peak loads, and ideally CPU utilization should be less than 25%, so we recommend quad CPU configurations for such customers.

This analysis is based on typical configurations, but the system supports creation of an unlimited number of business rules, each of which runs in the foreground or as a scheduled background task to call external scripts, generate emails, update related records and carry out other operations necessary to automate the business. The overhead from such processes can be significant, so for very large companies with highly sophisticated business rules, we support the IBM P-Series with up to 64 processors. Although few enterprises will actually need more than a couple of machines configured with dual Intel X5560 CPU's, support for the P-Series guarantees that plenty of headroom is available if the need arises.

Another aspect of scalability is the size of knowledgebase that can be supported while maintaining good performance. For common operations such as finding records containing certain text and editing them, scalability is almost unlimited.

From usage analysis on our ASP servers, we know that about 50% of CPU resources are spent submitting new records such as records and emails or editing existing ones, 20% is spent generating charts/reports, 15% is spent searching for information and 15% is spent on miscellaneous tasks such as changing tabs, logging in, viewing history information, creating saved searches, changing views, etc.

Conclusions

As detailed in figures 2a and 2b, for systems with two or more CPU cores, throughput increases by a factor of 1.5 to 1.9 each time that the number of active CPU cores is doubled. This is within 25% of the theoretical maximum scalability of 2 and indicates that the system is free of bottlenecks.

As detailed in figures 3a, 3b, and 5, testing with over 10,000,000 records and 100,000 active users demonstrates that the number of records and the number of concurrent users does not place a measurable load on the system. Performance is only affected by the number of records that are actually being edited at any one time.

Subject to the points raised in the Performance Analysis section, Agiloft can handle creation of over 3,800 records per minute on a system configured with dual E5530 CPU's, and can handle creation of over 1,320 records per minute on a system configured with dual Xeon E5440 CPU's, regardless of the number of logged in users or existing records.

For very high load systems, capable of handling over 3,000 records per minute, we therefore recommend dual i7 CPU's, such as E5530's coupled with SSD hard drives. For high availability, we recommend use of a hot-swap redundant server with replication and failover provided by DRDB and heartbeat.

About Agiloft

As the global leader in agile contract lifecycle management (CLM) software, Agiloft provides significant savings in purchasing, enables more efficient legal operations, and accelerates sales cycles, all while drastically lowering compliance risk. Agiloft's adaptable no-code platform ensures rapid deployment and a fully extensible system. Using contracts as the core system of commercial record, Agiloft's CLM software leverages AI to improve contract management for legal departments, procurement, and sales operations. For more information, visit https://www.agiloft.com.