



Agiloft Scalability and Redundancy

Table of Contents

3	Introduction
3	Performance on Hosted Server
	Figure 1: Real World Performance: CPU load averages for 30 days
3	Benchmarks
	System configuration used for benchmarks
	Figure 2: New records per minute on Intel Xeon E312xx
	Figure 3: Effect of the number of records on database search performance
	Figure 4: Effect of the number of records on file search performance
	Figure 5: Number of logins per minute to the End User Interface
	Figure 6: New logins per minute to the Power User interface
	Figure 7: Effect of the number of logged in users
5	System Requirements
5	Load Balancing
	Separate database and EJB servers
	Multiple database servers
	Multiple application servers
6	Server Redundancy
6	Performance Analysis
7	Conclusions

Introduction

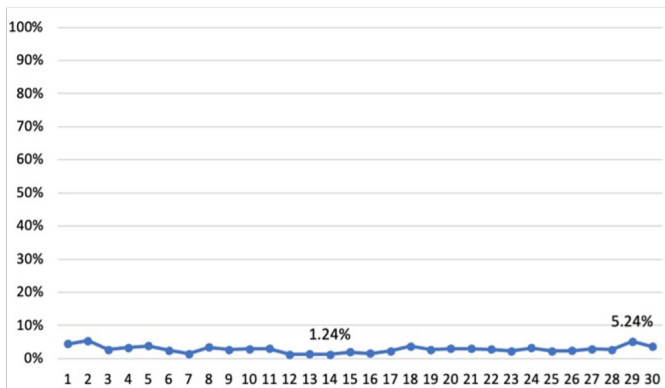
This paper reviews Agiloft scalability from the perspective of actual use on a hosted server and benchmarks on standard hardware. This analysis is followed by a discussion of the load that is placed on the system by typical users and a summary.

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: CPU load averages for 30 days



Comments:

These numbers were obtained from a server configured with a 12 core Intel Xeon E312xx processor and 192G RAM. The Agiloft implementation on this server has one of the largest databases and runs several very resource-intensive tasks every day.

Benchmarks

System configuration used for benchmarks

Database: MySQL

Hardware: 16 core Xeon E312xx @ 2.30GHz, 80G RAM

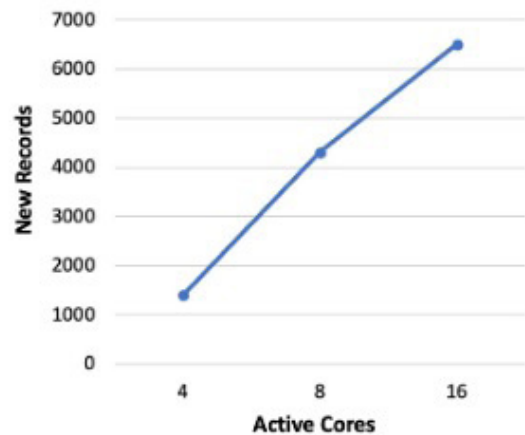
OS: Red Hat Linux

Agiloft Release: Release 23

Comments:

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 2: New records per minute on Intel Xeon E312xx



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 2, the number of records in the database does not have a measurable effect on performance, so a system with dual Intel Xeon E312xx processors (16 cores) can generate over 390,000 records per hour until it runs out of disk space.

Figure 3: 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.

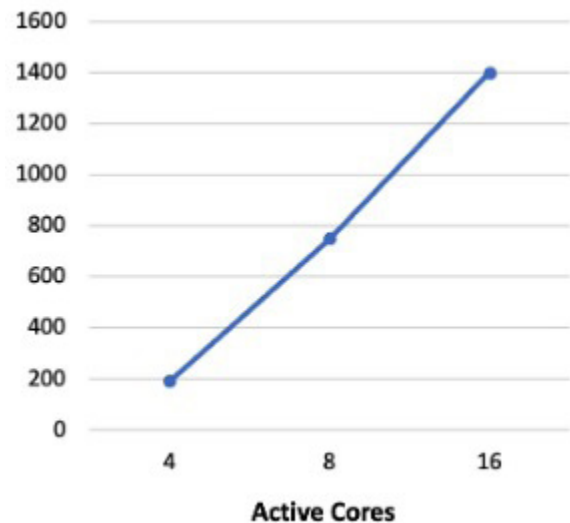
Figure 4: Effect of the number of records on file search performance

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.

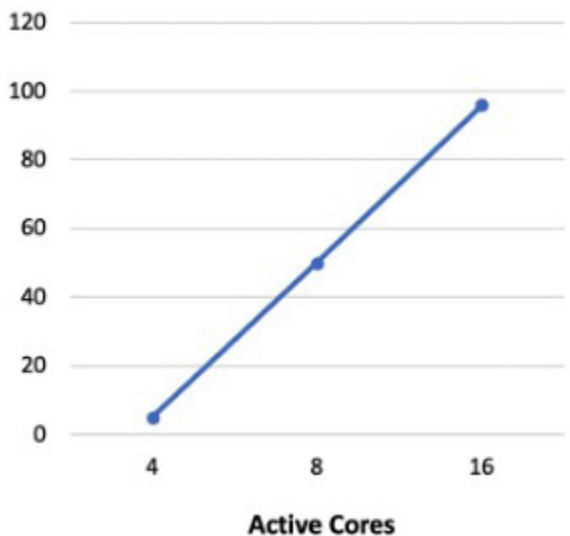
Figure 5: Number of logins per minute to the End User Interface



Comments:

These numbers reflect the number of unique users that may login to the Agiloft end user interface per minute. As detailed in Figure 7, 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 6: New logins per minute to the Power User interface



Comments:

These numbers reflect the number of unique users that may login to the Agiloft power user interface per minute. As detailed in Figure 7, 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 7: 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 6,000 users were creating new records at the rate of one per user per hour, there would not be a significant performance impact. 6,000 users who were each attempting to create new records at the rate of one per user per minute would bring a 16-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 WildFly. 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.

The minimum system requirements are easily surpassed by any modern server configuration. Agiloft can operate on a server with only 8G of RAM. If very little time is spent on I/O, then 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 for good performance.

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.

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 1Gbps connection. In practice, any performance gain from CPU utilization is often completely offset 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.

Server Redundancy

For high availability configurations in on-premises installations, 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.

This methodology can also be used if the secondary machine is located at a different facility. The use of DRDB proxy is recommended in this case.

Performance Analysis

System resources are only used when the user 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 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

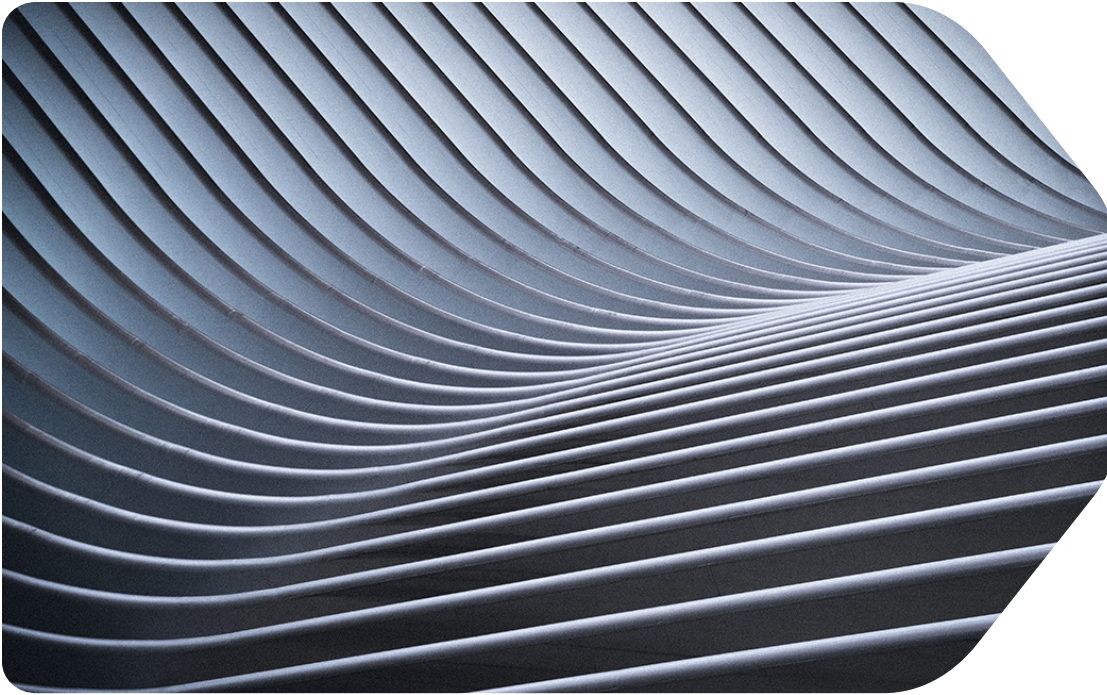
per day, we can expect a sales staff with 1,000 full time agents to generate or edit 8,000 records per day, significantly less than our entry level system can handle in half an hour. Of course, it is necessary to add a safety factor for sudden peak loads, and ideally CPU utilization should be less than 25%, so we recommend 8-core 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, or carry out other operations necessary to automate the business. The overhead from such processes can be significant. However, the current hardware configuration of Agiloft's hosted offering is more than capable to support any enterprise implementation.

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 Figure 2 and in Figure 3, for systems with four 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 Figure 4, Figure 5, and Figure 6, 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 being edited at any one time.

Subject to the points raised in the Performance Analysis section, Agiloft can handle creation of over 6,500 records per minute on a system configured with 16 core Intel Xeon E312xx, regardless of the number of logged in users or existing records.

For high availability, we recommend use of a hot-swap redundant server with replication and failover provided by DRDB and heartbeat.

About Agiloft, Inc.

As the global leader in agile contract lifecycle management (CLM) software, Agiloft is trusted to provide significant savings in purchasing, enable more efficient legal operations, and accelerate 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. Visit www.agiloft.com for more.