Benchmark Test Results: Instec Quicksolver Running on Microsoft SQL Server

Benchmark testing demonstrates excellent results for Instec Quicksolver 3.7 running on Microsoft SQL Server 2012, Windows Server 2012, and Windows 8 client computers

Abstract

Instec and Microsoft worked together to test the performance and scalability of Instec Quicksolver 3.7 insurance policy administration software running on Microsoft SQL Server 2012 data management software, the Windows Server 2012 operating system, and Windows 8 client computers. The benchmark tests were designed to simulate real-world conditions and used a mix of activities typical for a large, tier-one insurance carrier.

The results of the tests were impressive, confirming the performance and scalability of Quicksolver on SQL Server and Windows Server. This report provides the results of the tests, in addition to details about the test environment and the test methodology used.

Table of Contents

1. Introduction **1**

1.1 Introducing Instec Quicksolver *1*

1.2 Benefits of SQL Server 2012 ..... *1*

2. Overview of the Benchmark Tests **2**

3. Benchmark Test Results **2**

3.1 Results of Scale-Up Tests *2*

3.2 Results of Scale-Out Tests *3*

3.3 Summary of Conclusions *4*

4. Benchmark Test Environment **4**

4.1 Software Details *5*

4.2 Hardware Details *6*

5. Benchmark Test Procedures **6**

5.1 Data Sampling Procedures *7*

5.2 Test Activities *7*

5.2.1 Data Entry 7

5.2.2 Ratings 8

5.2.3 Forms 9

5.2.4 Worksheets 10

5.3 Baseline Tests *11*

5.3.1 Baseline Test Mixes 11

5.3.2 Time Delays 12

6. Additional Observations **13**

7. Summary **13**

8. Appendix: Determining Hardware Requirements **14**

9. Appendix: Benefits of SQL Server 2012 ..... **16**

10. Appendix: Benefits of Quicksolver **17**

11. Additional Information **18**

1. Introduction

Instec—a leading provider of software and services to the commercial property/casualty insurance industry—and Microsoft worked together to test the performance and scalability of Quicksolver 3.7 running on Microsoft SQL Server 2012 data management software, the Windows Server 2012 operating system, and client computers running Windows 8. The tests used workloads typical for tier-one insurance carriers.

> "This latest release of Quicksolver 3.1 is further evidence that Instec is a leading player in the insurance value chain by consistently delivering well-engineered, practical software and services."
>
> *Colin Cole*
> *Chief insurance technology strategist*
> *Microsoft*

The benchmark tests were conducted at the Microsoft Platform Adoption Center (PAC) in Redmond, Washington, in February of 2013. They built on previous work conducted by Instec and Microsoft in 2009 at the Microsoft Test Center in Chicago, Illinois, using Quicksolver 3.0, SQL Server 2005, and Windows Server 2003 R2 Enterprise.

The tests used industry-standard servers to simulate a real-world production environment and scaled both up (with the number of cores ranging from 2 to 24 and the amount of memory configured from 2 GB to 32 GB) and out (with the number of virtual machines ranging from one to eight).

The test results were very impressive, confirming that the solution can meet the performance and scalability requirements of very large tier-one insurance carriers. This report provides the results of the performance tests, in addition to details about the test environment and the test methodology used.

1.1 Introducing Instec Quicksolver

Instec's flagship technology, Quicksolver, provides full policy lifecycle management. The Quicksolver solution is a complete property and casualty (P&C) insurance rating and policy administration system (PAS) that supports multi-state, multi-location, bureau-specific, and client-specific policy processing for all 50 states. The Quicksolver solution makes it possible for clients to effectively manage their businesses, from first quote to last endorsement, for commercial and specialty lines of insurance. The Quicksolver solution is noted for its powerful custom solutions that enable clients to drive new revenue and achieve profitable growth.

For more detail about Instec Quicksolver, see **Section 10. The Benefits of Quicksolver**.

1.2 Benefits of SQL Server 2012

The underlying core of an insurance solution is the data collected; this data is maintained in an enterprise database such as SQL Server 2012. Running on the Windows Server operating system,

SQL Server 2012 provides a reliable, high performing, and scalable foundation for Instec Quicksolver.

For more detail about the advantages SQL Server brings to the solution, see **Section 9. The Benefits of SQL Server**.

2. Overview of the Benchmark Tests

This section provides an overview of the benchmark test procedure and test results. For more detail, see the sections that follow.

For the previous tests in 2009, engineers from Instec developed a set of test scenarios comprising typical user activities: entering data; rating; selecting, generating, and previewing forms; and generating and previewing worksheets. These scenarios were used to create a baseline server load for each test category. This baseline was then used in a sample mix of categories to approximate a typical installation. The sample mixes were then measured against a series of an increasing number of virtual machines and virtual users to gauge the scalability of Quicksolver 3.7.

The mixed tests were designed to approximate a group of users concurrently and actively using Quicksolver. Three test scenarios were designed using a mix of these activities to represent typical, light, and heavy workloads. These test scenarios were run against a variety of hardware configurations with varying user loads. The team simulated load balancing by using multiple agents, each mapped to different application server.

For the tests run in 2013 in Redmond, the baseline server loads were first tested and compared to the previous tests. The primary focus was then to test the performance of Quicksolver 3.7 at processor saturation for various configurations. Test results are reported for the typical workload scenario, to best simulate real-world conditions. Based on recommendations from Microsoft, processors were considered saturated when the length of the processor queue averaged between two and three threads per core.

3. Benchmark Test Results

The test results showed that when the servers were configured to optimize memory access, the performance was linear and predictable up to the point of CPU saturation. Beyond the point of CPU saturation, the application continued to function, but some functions not related to the user interface did experience longer response times. Note that the Quicksolver 3.7 user interface functions run at a higher priority than other processes.

3.1 Results of Scale-Up Tests

Figure 1 shows an overview of the “scaling up” tests for a typical test scenario, using average-sized policies with one to three locations (see **Section 5. Benchmark Test Procedures** for a detailed

description of the workloads). The graph shows the number of tests completed by core and by user.

Note that the leveling off in the number of tests completed shows that even when the user load exceeds the recommendations, the performance of Quicksolver 3.7 remains stable and predictable.

These results show that Quicksolver 3.7 can scale up in number of cores to the physical node boundary; beyond that, it should be scaled out with additional physical servers or with additional virtual machines.

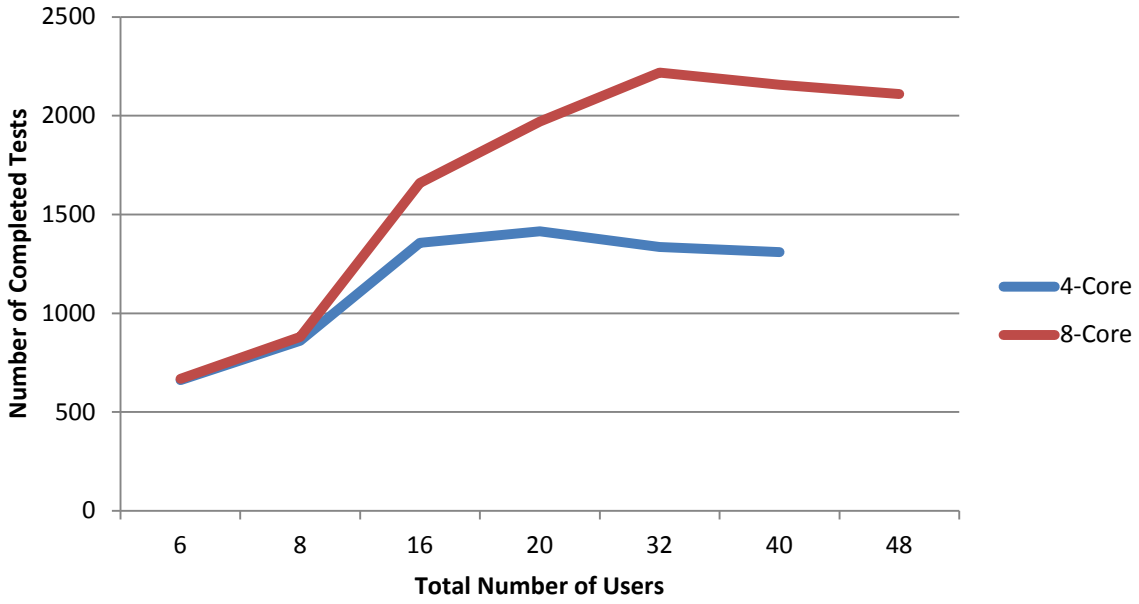

Figure 1. Scaling up test results

3.2 Results of Scale-Out Tests

Figure 2 shows an overview of the "scaling out" tests for a typical test scenario, also using average-sized policies with one to three locations (see **Section 5. Benchmark Test Procedures** for a detailed description of the workloads). The graph shows the number of tests completed at CPU saturation.

Note that the tests show a completely linear progression, with multiple virtual machines configured with from two to eight cores.

These results show that Quicksolver 3.7 can dependably and predictably scale out when running on SQL Server 2012 and Windows Server 2012.

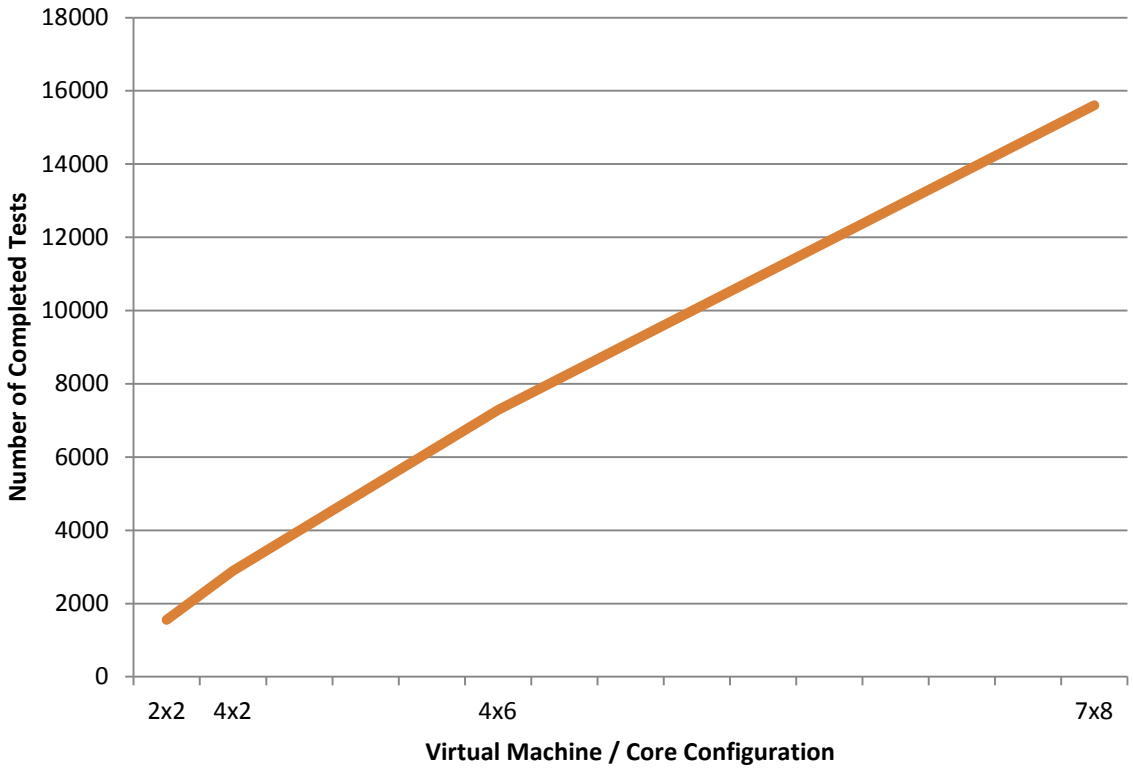

Figure 2. Scaling out test results

3.3 Summary of Conclusions

The main goals of the benchmark tests were met. The test results confirmed that:

- Instec Quicksolver 3.7 supports SQL Server 2012, Windows Server 2012, and client computers running Windows 8.
- The performance of Quicksolver 3.7 running on Microsoft software is linear when scaled up to a limit; beyond this limit, performance is still stable and predictable.
- The performance of Quicksolver 3.7 running on Microsoft software is linear and predictable when scaled out.

4. Benchmark Test Environment

The sections that follow provide more detail about the test environment.

The benchmark tests used up to eight virtual machines on a virtual machine host server, a database server, and a file server. Eight workstations were used as clients; two of these were used as test controllers, which coordinated the agents and collected the results.

Figure 3 shows the benchmark test environment.

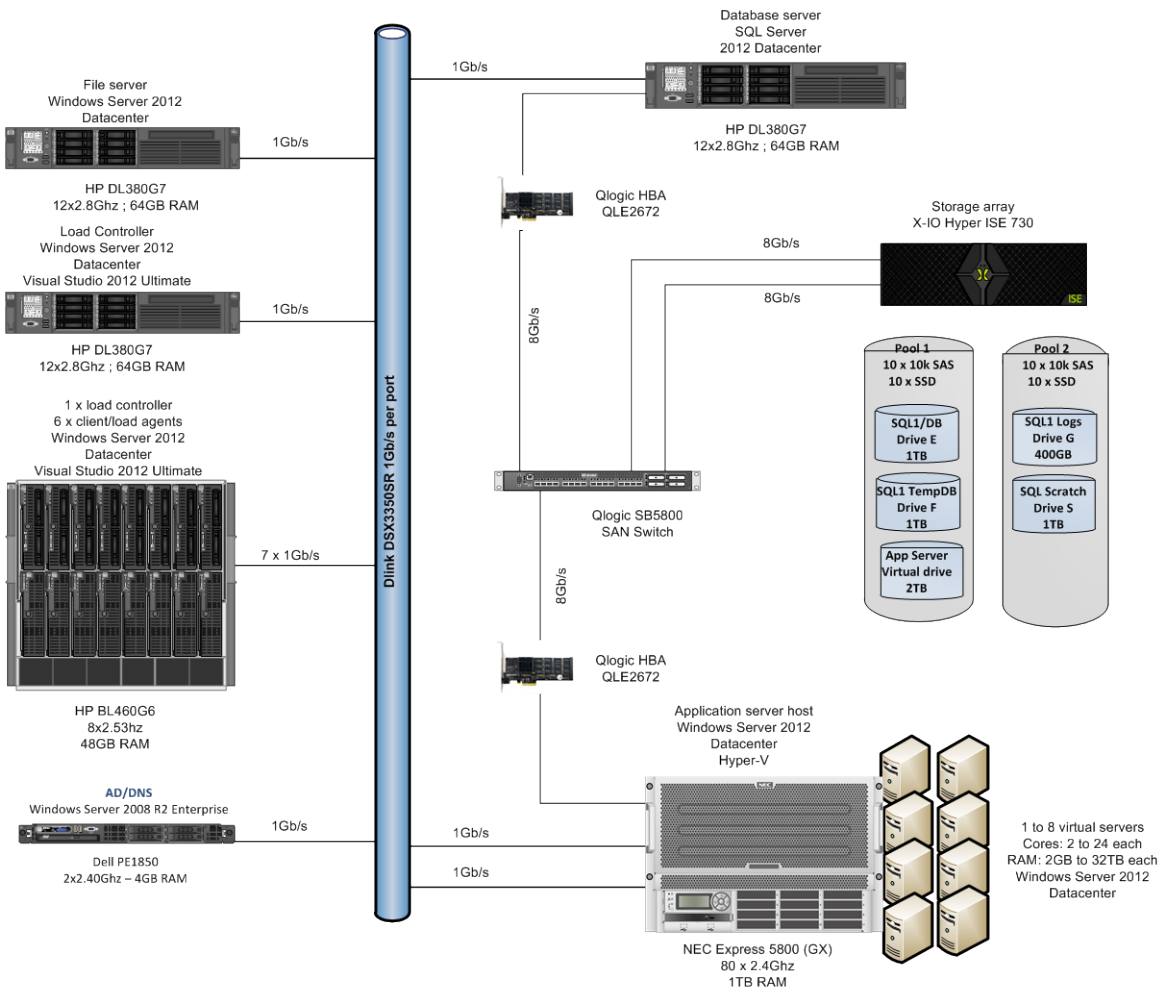

Figure 3. Benchmark test environment

4.1 Software Details

The team used the following software versions in the tests:

- SQL Server 2012 (database server)
- Quicksolver 3.7
- Windows Server 2012 Datacenter (all servers except the domain controller)
- Windows Server 2008 R2 (domain controller)
- Hyper-V (enabled on the virtual machine host server)
- Windows 8 (client computers)
- Visual Studio 2012 Ultimate

4.2 Hardware Details

Table 1 provides details for the hardware used in the benchmark testing.

Table 1. Hardware details

Server role	Make and model	Number/type of CPUs	Cache	Number of cores	RAM
Database server	HP ProLiant DL380 G7	2 Intel Xeon processor X5660 (2.80 GHz)	12 MB	12	64 GB
Virtual machine host	NEC Express5800/ A1080a-E	8 Intel Xeon processor E7-4870 (2.40 GHz)	30 MB	80	1 TB
File server	HP ProLiant DL380 G7	2 Intel Xeon processor X5660 (2.80 GHz)	12 MB	12	64 GB
Controllers (2)	HP ProLiant DL380 G7	2 Intel Xeon processor X5660 (2.80 GHz)	12 MB	12	64 GB
	HP ProLiant BL460 G6	2 Intel Xeon processor E5540 (2.53 GHz)	8 MB	8	48 GB
Agents (6)	HP ProLiant BL460 G6	2 Intel Xeon processor E5540 (2.53 GHz)	8 MB	8	48 GB
Domain controller/Active Directory Domain Services	Dell PowerEdge 1850	2 Intel Xeon processor (3.40 GHz)	2 MB	2	4 GB

5. Benchmark Test Procedures

To test the performance and scalability of Quicksolver 3.7 on SQL Server 2012 and Windows Server 2012, the team scaled the system both up and out:

- **Scaling up**.
  Tests were performed with the number of core ranging from 2 to 24 and with memory configured from 2 GB to 32 GB.
- **Scaling out.**
  The number of virtual machines was scaled from one to eight.

The team designed test scenarios to simulate the workloads of a typical large insurance carrier. Each test scenario used a mix of four types of activities and ran for 20 minutes, using average-sized policies with between one and three locations.

The four types of activities were first used to create a baseline for the server load; these activities are those typically performed by a group of concurrent, active Quicksolver users. The baseline results were used to design sample mixes (light, typical, and heavy) of activities to approximate real-world workloads. The sample mixes were then measured against a series of increasing hardware and virtual users to test the scalability of the system.

During the tests, the team used the Microsoft reliability and performance monitor (Perfmon) to collect performance statistics for the servers and test agents through the test controllers. The team

added custom Quicksolver performance counters to monitor application-specific data, such as number of open policies, the number of policies rating, and the number of active policy result channels. A test agent controller was used to launch the test agents and consolidate the Perfmon statistics and custom counters into a local SQL Server database. The team then analyzed the data that was collected.

5.1 Data Sampling Procedures

All tests were run for 20 minutes. A warm-up period of between one and five minutes was used to gradually ramp up the number of users. Similarly, a three to five minute cool-down period was used to close all test policies and end-channel communications. Performance statistics were not collected during the warm-up and cool-down periods.

5.2 Test Activities

The team first performed a set of baseline tests to measure the behavior of Quicksolver during four activities typically performed by Quicksolver users:

- Data entry
- Rating
- Forms (selection, generation, and preview)
- Worksheets (generation and preview)

5.2.1 Data Entry

The data entry test simulates the creation of a typical, one-location New Business Owner Policy quote. The quote is the same for all users: the rating state, producer, rating company, class code, deductibles, and limits are all the same. Only the insured name is changed to include a time stamp to differentiate the quotes.

The data entry test uses the following operations:

1. Begin new quote.
   This step opens a new instance of a policy document.
2. Enter data.
   The quote includes enough data entry for a successful rating, including:
   - Policy information
   - Insured name and address
   - Producer
   - Policy-level required coverage
   - Location and state
   - Building coverage
   - Personal-property coverage
   - Liability coverage

3. Save the quote.
4. Close the quote.

This data entry test generates the most activity (CPU and input/output [I/O]) in the server at the end of the activity (steps 3 and 4).

Each complete data entry test took about four minutes. Variations between test runs are due mainly to the bracketed random wait times used in the test.

5.2.1.1. Measuring rating tests

The metric used to measure data entry is the maximum number of users that the server can service before the users experience a slowdown in data-entry response time; this occurs when the processor queue length average is greater than four.

5.2.1.2 Effect of the test environment

In the typical Quicksolver user interface, a user entering data can visually identify a data-entry field. The user sees fields in a screen context and might see a label next to or above a field to identify it. However, the client and server exchange field-identification information at an abstract level. The automated test process, acting as a client, cannot readily discern the identity of any particular field from the dynamic data returned to it by the server.

To make automated field identification possible, the team added an *Extended Field Description* feature. This feature provides additional information for every field, which can be used to uniquely identify the field. The baseline tests turn this feature on whenever field identification is required. The overhead of this feature is negligible; production response is likely to be slightly better than the test results.

5.2.2 Ratings

The rating test, in addition to the forms and worksheets tests, used pre-entered Business Owners, Commercial Package, Commercial Auto, and Workers Compensation insurance policies. Test policies were evenly distributed among the test agents. Each test agent initiated testing by building a private queue of policies from those allocated to it.

The rating baseline test consisted of opening a policy, rating the policy, and saving the result:

1. Open a policy.
   This step opens a new instance of a policy document from the top of the queue.
2. Rate the policy.
   The policy is rated and a *Completed Rating* dialog box is sent to the client computer. The client computer dismisses the dialog after checking for a successful rate process.
3. Save the policy.
   The policy is persisted to the file server.
4. Close the policy.
   The policy is placed back at the end of the policy queue for the next test.

This rating test is I/O intensive at the beginning and at the end of the activity (steps 1 and 4), and it is CPU intensive during the rating process (step 2).

The average time required for each complete unit test ranged from 6.4 seconds to 13.7 seconds, mainly depending on the number of locations in the policy (one to three locations, respectively).

5.2.2.1 Measuring rating tests

The metric used to measure rating is the maximum number of users that the server can service before the users experience a slowdown in data-entry response time; this occurs when the processor queue length average is greater than four.

5.2.2.2 Effect of the test environment

Normally, a user does not open, rate, and close a policy without also entering data or reviewing worksheets. However, to keep the rating activity isolated, the test incurred the overhead of opening and closing the policy. When the rating test is used in a mix with other tests that are also opening and closing the policy, the mix includes many more opens and closes than would normally occur in a production environment.

5.2.3 Forms

The forms selection, generation, and preview test used pre-entered Business Owners, Commercial Package, Commercial Auto, and Workers Compensation insurance policies. Test policies were evenly distributed among the test agents. Each test agent initiated testing by building a private queue of policies from those allocated to it.

The forms baseline test consisted of the following steps:

1. Open a policy.
   This step opens a new instance of a policy document from the top of the queue.
2. Select forms.
   This activity determines which forms and form editions were to be attached to the policy based on the underlying coverages. This step is CPU intensive.
3. Generate forms.
   PDF versions of forms are generated from form templates, with data that is merged onto the form from underlying policy data. PDF forms are persisted in the SQL Server database for the user to retrieve. This step is mainly I/O intensive, with short bursts of CPU demand.
4. Download forms.
   PDF versions of the files are requested from the repository and downloaded to the local workstation (the test agent). This step is I/O intensive.
5. Close the policy.
   The policy is placed back at the end of the policy queue for the next test.

This forms test is I/O intensive when forms are being downloaded (step 4) and also includes bursts of CPU-intensive activity when forms are generated (step 3).

The average time required for each complete unit test ranged from 19 seconds to 35 seconds, mainly depending on the number of forms in the policy, which depends on the number of locations, states, and coverages.

5.2.3.1 Measuring rating tests

The metric used to measure forms selection, generation, and preview is the maximum number of users that the server can service before the users experience a slowdown in data-entry response time; this occurs when processor queue length average is greater than four.

5.2.3.2 Effect of the test environment

This test opens and closes the policy for each set of forms requested. When this test is used in a mix with other tests that are also opening and closing the policy, the mix includes many more opens and closes than would normally occur in a production environment.

After a set of forms is downloaded to the workstation, a typical user would open the PDF and review and/or print the forms. These functions would take place entirely on the local workstation (with the exception of printing, which might involve some network traffic). This test does not include wait time for viewing. As soon as the PDF file is downloaded and the policy is closed, another test is initiated. For this reason, the server is kept busy with forms requests more than would normally occur in a production installation with the same test parameters (such as number of users).

5.2.4 Worksheets

The worksheets generation and preview test used pre-entered Business Owners, Commercial Package, Commercial Auto, and Workers Compensation insurance policies. Test policies were evenly distributed among the test agents. Each test agent initiated testing by building a private queue of policies from those allocated to it.

The worksheets generation and preview test used the following operations:

1. Open a policy.
   This step opens a new instance of a policy document from the top of the queue.
2. Create worksheets.
   All available worksheets are requested. For a Business Owners policy, this includes these worksheets:
   - Detailed worksheet
   - Tax, surcharge, and fees detailed worksheet
   - Condensed worksheet
   - Policy totals worksheet
   - Individual risk premium modification (IRPM) worksheet
   - Policy notes
   - Statistical coding worksheet
   - Warning messages worksheet

3. Generate forms.
   PDF forms are generated dynamically from underlying policy data. PDF forms are persisted in the SQL Server database for subsequent retrieval by the user. This step is CPU intensive.
4. Download forms.
   PDF files are requested from the repository and downloaded to the local workstation (the test agent). This step is I/O intensive.
5. Close the quote.

This worksheet test is primarily CPU intensive; worksheets are relatively small in comparison to forms, and most of the work to generate the PDF is done in memory.

The average time required for each complete unit test ranged from 6 seconds to 10 seconds, mainly depending on the number of locations and coverages.

5.2.4.1 Measuring worksheets tests

The metric used to measure worksheets generation and preview is the maximum number of users that the server can service before the users experience a slowdown in data-entry response time; this occurs when the processor queue length average is greater than four.

5.2.4.2 Effect of the test environment

This test opens and closes the policy for each set of worksheets requested. When this test is used in a mix with other tests that are also opening and closing the policy, the mix includes many more opens and closes than would normally occur in a production environment.

After a set of worksheets is downloaded to the workstation, a typical user would open the PDF and review and/or print the worksheets. These functions would take place entirely on the local workstation (with the exception of printing, which might involve some network traffic). This test does not include wait time for viewing. As soon as the PDF file is downloaded and the policy is closed, another test is initiated. For this reason, the server is kept busy with worksheet requests more than would normally occur in a production installation with the same test parameters (such as number of users).

5.3 Baseline Tests

The team conducted baseline tests for the four types of activities. These baseline tests were designed to emulate a typical user's interaction with the user interface. Note that an automated process replaced the Quicksolver user interface in the tests; this automated process communicated with the Quicksolver services in the same way that the user interface does. The automated interface and the Quicksolver user interface could communicate with the Quicksolver services at the same time. In the tests, the Quicksolver user interface was used to gauge the responsiveness of the server to user interface requests.

5.3.1 Baseline Test Mixes

The baseline tests combined various percentages of the different types of tests into three mix

scenarios. These scenarios characterize the workloads of a typical large insurer, as the types of quotes, policies, and tasks shift during the course of a day, month, and year: at normal-use periods, there might be more data entry than rating, while during renewal processing, there might be a relatively small amount of data entry but a lot of rating and forms generation. Table 2 describes the test mixes used.

Table 2. Test mixes

Scenario	Data entry	Forms	Worksheets	Ratings
Typical mix	45 percent	15 percent	20 percent	20 percent
Light mix	75 percent	5 percent	10 percent	10 percent
Heavy mix	30 percent	20 percent	20 percent	30 percent

Because each baseline test took a different amount of time to complete, the percentage of the baseline tests was based on the number of tests started; by the end of test, the given percentage of each of the baseline tests was approximately equal to the number of tests completed during the test sampling.

5.3.2 Time Delays

A time delay, or "think time," was built into the data entry baseline tests to simulate human response time: each client-side submission to the server is delayed to approximate the time a human takes to read the screen, position the cursor, select or type the desired value, and press enter or tab to move to the next field. Note that the tests did not include other time delays, such as when a user is called away from the process to answer their phone or converse with a co-worker.

Time delays were categorized as short, medium, or long. Each time-delay category was defined with a range of values from shortest delay to longest delay. Each use of a delay (a delay instance) is computed using a randomly generated value within the category's range of values. For example, a short delay can range from one to three seconds. Note that time delays were computed in milliseconds (ms). Table 3 shows the time delays used in the tests.

Table 3. Test scenarios

Delay category	Shortest delay	Longest delay	Example of use
Short	1 second	3 seconds	Click **OK**. Select a radio-button option.
Medium	2 seconds	4 seconds	Use a drop-down list. Type an entry in a text field.
Long	4 seconds	8 seconds	Move to a new screen. Type additional information in a dialog box.

6. Additional Observations

A benefit of testing in the Microsoft Platform Adoption Center is the ability to run a variety of configurations and large-scale tests in a controlled environment. The Instec engineers were able to make the following observations about Quicksolver running on SQL Server 2012 and Windows Server 2012:

- To optimize performance, virtual machines should be sized as a multiple of the physical server's NUMA node size.
- Scaling out is more efficient than scaling up.
- Maximum throughput (tests completed) occurs when the configuration makes one core available for every four to six concurrent users.
- Because of the fixed number of file handles per server, each virtual machine can accommodate no more than 400 users.
- More servers with fewer cores appears to be more efficient than an equal number of cores on fewer servers. For example, there was slightly better performance in a 4 by 6 configuration versus a 3 by 8 configuration.

7. Summary

The benchmark test results confirm that SQL Server 2012 and Windows Server 2012 are an excellent choice to run Quicksolver 3.7. Instec and Microsoft plan to continue to work together on additional tests, including long-running tests of seven or more days and performance and scalability tests of the next Quicksolver release.

8. Appendix: Determining Hardware Requirements

The performance model and benchmark testing represent a starting point for determining a customer's actual installation requirements. Many variables can affect performance at an installation.

Some customers fit comfortably within the model's boundaries and do not require investigation to determine their actual hardware requirements. For example, a customer might have 30 insurance agents who use Quicksolver occasionally, each processing several insurance transactions throughout the day. There is never a peak period where all agents are performing intensive processing. In this case, one server can adequately support the customer's performance requirements.

Other customers might have more complex requirements, and it might be necessary to construct test scenarios that more accurately reflect the proposed processing.

The following list provides questions to ask and items to consider when planning the installation:

- Are there specific hardware requirements (types of processors and/or operating systems)?
- How many users are there? What types of users?
    - Highly active users? Consider the number of quotes/policies/transactions processed per day.
    - Frequent use or infrequent/inquiry-only use?
- Are there specific work patterns?
    - Is the workload heavy in the morning? Or heavy at the end of the day?
    - Are the users scattered across time zones?
- What is the policy/quote/transaction mix? This might best be determined by looking at annual counts of:
    - Number of quotes
    - Number of issues
    - Number of endorsements
    - Number of renewals
- What types of policies are being processed?
    - Business Owners
    - Package
    - Commercial Automobile
    - Workers Compensation
    - Other or specialty lines
- What is the size of the majority of policies?

- Are they single state or multi-state?
- How many locations are on a typical policy?
- How many covered autos are on a typical policy?
- For workers compensation, how many class codes are on a typical policy?

- Are there any unique processing cycles?
  - Is the processing even throughout the year?
  - Is there heavy month-end processing?
  - Does the entire book of business renew on one date or within a short period?
  - Are there other peak periods of processing?

- Other considerations:
  - Customers should decide on the level of performance required. For example, it might be acceptable for an occasional slow response during peak periods. In that case, more users per server can be supported than the stated recommendations.
  - Is there any automated processing? For example, are policies printed via a batch process in the evening when there is no user activity? Is there an automated process that initiates quotes or transactions?

9. Appendix: Benefits of SQL Server 2012

SQL Server 2012 provides customers with many benefits, including:

- **Required nines uptime availability and data protection**.
  Customers can protect their mission-critical databases from downtime and data loss with SQL Server AlwaysOn,[1, 2, 3] the new integrated high availability and disaster recovery solution.
- **Faster deployment**.
  SQL Server database administrators (DBAs) can typically install and configure new database servers in 1.5 hours, while the largest competitor's DBAs can take 6 hours or more.[4]
- **Lower hardware costs**.
  SQL Server can run on standard commodity server hardware, which can dramatically lower the total cost of ownership (TCO).
- **Lower software costs**.
  The list price of SQL Server is a third of the largest competitors;[5] in addition, SQL Server includes major database-related features, such as high availability; remote disaster recovery; partitioning; data compression; transparent data encryption; spatial; master data management; complex event processing; Extract, Transform, and Load (ETL); online analytical processing (OLAP); data mining; reporting services; and self-service BI tools. Competitors' licensing models add costs for options and add-ins.[6]
- **Simpler systems management and lower staffing costs**.
  SQL Server DBAs can typically manage four times as many physical databases as a competitor's DBAs, leading to an estimated annual savings of $5,779 in administrative costs per database, a 460 percent difference in annual cost of administration per database.[7]
- **Fewer security vulnerabilities**.
  Since 2002, SQL Server has recorded the fewest reported vulnerabilities of any of the major database platforms as compiled by the National Institute of Standards and Technology.[8]

By using SQL Server 2012, Instec customers can save with reduced hardware, administration, and support fees, which translate into substantially lower costs over the life of the system.

For more information about SQL Server, see **www.microsoft.com/en-us/sqlserver**.

[1] http://www.microsoft.com/sqlserver/en/us/product-info/why-sql-server.aspx
[2] http://www.microsoft.com/casestudies/Microsoft-SQL-Server-2008-R2-Enterprise/Stratus-Technologies/Protect-your-mission-critical-databases-from-downtime-and-data-loss-with-six-nines-uptime-availability/4000007136
[3] http://download.microsoft.com/download/D/2/0/D20E1C5F-72EA-4505-9F26-FEF9550EFD44/Microsoft%20SQL%20Server%20AlwaysOn%20Solutions%20Guide%20for%20High%20Availability%20and%20Disaster%20Recovery.docx
[4] http://www.alinean.com/PDFs/Microsoft_SQL_Server_and_Oracle-Alinean_TCA_Study_2010.pdf
[5] http://www.microsoft.com/sqlserver/en/us/tools/cost-savings-calclator.aspx
[6] http://www.microsoft.com/sqlserver/en/us/product-info/competitor-compare.aspx
[7] http://www.microsoft.com/sqlserver/en/us/product-info/why-sql-server.aspx
[8] http://itic-corp.com/blog/2010/09/sql-server-most-secure-database-oracle-least-secure-database-since-2002

10. Appendix: Benefits of Quicksolver

The Quicksolver solution is a service-oriented architecture (SOA) incorporating Microsoft .NET services to efficiently manage thousands of changes annually, while preserving the integrity of client configurations. Through five generations of technology change, Instec has met its "No Legacy" pledge by moving all client investments forward to each new platform without reinvestment or recoding.

The Quicksolver solution separates application functions into self-contained services based on SOA and .NET standards. Each service runs as an independent unit, free of dependencies on information storage or other business services.

The Quicksolver solution's cohesive application software architecture takes advantage of:

- Policy rating and document management services
- SQL Server database
- Business rules and configurations for commercial and specialty lines for all states
- Integration services for external line-of-business applications
- Electronic software delivery and change-management services

The Quicksolver solution is an object-oriented n-tier architecture that isolates all bureau-supplied business rules/changes from client-specific rules/changes or configuration settings. The Quicksolver solution publishes each change digitally, and software updates are integrated without interruption with electronic software delivery services. Each year, month, and day, client-independent insurance product and program changes are collated with thousands of annual, date-sensitive, multi-state rate/rule/form/statistical data collection industry updates and state compliance rule exceptions. Instec analyzes thousands of bureau updates from the Insurance Services Office (ISO), National Council on Compensation Insurance (NCCI), and the Independent Workers Compensation bureaus. Instec delivers revisions in granular monthly updates, so clients get the relevant revisions while preserving the integrity of extensions. Instec has been a business partner with all independent bureaus for more than 20 years and delivers over 3,000 revisions annually. The Quicksolver solution also includes a library of over 300,000 rates, rules, and statistics and more than 20,000 forms.

For more information about Quicksolver, see **www.instec-corp.com/quicksolver-property-casualty-product.html**.

11. Additional Information

The following references provide more information about Instec, Microsoft, Intel, X-IO, HP, and Qlogic:

- For more information about Instec, visit: **www.instec-corp.com**
- For more information about Microsoft, visit**:** **www.microsoft.com**
- For more information about Intel, visit: **www.intel.com**
- For more information about X-IO, visit: **www.xiostorage.com**
- For more information about NEC, visit: **www.nec.com**
- For more information about Qlogic, visit: **www.qlogic.com**