

An Oracle White Paper
June 2010

Maximizing Performance and Scalability of a Policy Automation Solution

Executive Overview

Most enterprises are now moving towards a modern IT architecture that is based on a service-oriented model. Key goals include ensuring that business policies are applied consistently across the organization, and are easier to update centrally as business needs change. Web services play an important role in the target architecture, but often concerns remain about how to match the performance of existing systems using this approach. This whitepaper explains how a well designed enterprise policy automation solution maximizes throughput, scalability and availability. It explains how batch processing, self-service guidance, and web services can be efficiently integrated into end-to-end business processes. This allows organizations to adopt the latest policy automation solutions available with confidence, knowing that they will meet and grow with their organization's needs.

Introduction: What is a Policy Automation Solution?

Policy automation involves taking business policies, and efficiently incorporating those policies into business processes to deliver accurate, consistent and auditable outcomes. The policies being automated may need to match particular legislation or regulations, or may simply need to reflect the latest internal company policies. For example, the policies may be designed to calculate payment amounts, make decisions on eligibility, or calculate risk ratings.

A policy automation solution needs to support modeling policies in an intuitive efficient manner. It also needs to make it easy to deploy modeled policies across the enterprise, using a high performance and scalable suite of policy automation components that integrate easily with both legacy and modern applications.

How Scalable does a Policy Automation Solution Need to Be?

To understand the typical load on a Policy Automation Solution, consider the example of administering benefit payments for citizens of a country with tens of millions of people.

When the government brings in a new benefit, many citizens will want to know if they are eligible for that benefit. An ideal solution to this requirement is a self-service application – based on the government’s benefit legislation – that allows citizens to complete a guided web-based interview to determine their eligibility. At its peak, this application may need to handle tens or even hundreds of thousands or more users each day.

If a citizen determines they are eligible, they are likely to want to file a claim for payment. Using a case management system as the front end to manage the data about the citizen’s claim, the policy automation solution needs to integrate cleanly with that CRM system to support the calculation of eligibility, and return the calculated payment amounts. Again, hundreds of thousands or even millions of calculations may need to be made daily.

Finally, consider the case of a change in benefit legislation, and the resulting need to recalculate the benefit amount that every eligible citizen currently receives. If 10% or more of citizens are eligible, millions of automated policy-based calculations will need to be processed, in a batch processing mode, in a short period of time.

How is a Policy Automation Solution Deployed?

As can be seen from the previous example, it is important that modeled policies can be easily deployed into a variety of systems:

- Web-based self-service applications
- With CRM systems or other applications, for responsive real-time decision making
- Into high-speed batch processing scenarios

Each of these has different technical requirements. In the case of Oracle Policy Automation, the following product components meet these needs:

- Oracle Web Determinations, for interactive policy-driven guidance
- Oracle Policy Automation Connector for Siebel, for integration with case management
- Oracle Determinations Server, for standards-based web service integration with other applications, middleware, or bespoke applications
- Oracle Determinations Engine Data Source Connector, for batch processing
- Oracle Policy Automation Connector for SAP Java Connector, for high performance integration with SAP applications

What Affects the Performance of a Policy Automation Solution?

When selecting a policy automation solution, it is important to understand what can affect performance of that solution. The key areas to consider are:

- Processing efficiency – e.g. is it optimized for modern CPU hardware?
- Impact of policy model size – e.g. is it designed to handle large or complex models?
- Data handling approach – e.g. is direct database connectivity avoided?
- Scalability – e.g. can standard web application scaling approaches be used?
- Batch processing capabilities – e.g. can it efficiently handle multiple decisions at once?

The approach of Oracle Policy Automation to each of these is considered in turn below.

Processing Efficiency

The engine embedded within Oracle Policy Automation implements an algorithm invented by Oracle engineers called *Linear Inferencing*. This algorithm has been specifically designed to take advantage of the large on-chip cache found in modern CPUs. Linear inferencing works by keeping in memory a single read-only dependency graph for the policy model, which is shared by all active sessions. Since this graph is almost always smaller than the available on-chip cache, this helps ensure that processing policy models is extremely fast.

Impact of Policy Model Size

Oracle Policy Automation enforces a modeling style that ensures every policy model can be quickly and consistently processed, regardless of what data is used. In this approach, there is no iteration or “rule-firing” cycle that takes place – a single pass through the rules is always sufficient to know what decisions can be reached. This ensures that the impact is minimized of increases in policy model size and complexity, and eliminates the need to go through a policy modeling optimization process prior to deployment. Because the rules are effectively always optimized, the result is faster processing, for every model size and complexity.

Data Handling Approach

Oracle Policy Automation's Determinations Server operates completely independently from any application database. That is, each web service call to Oracle Determinations Server passes all the data needed to make each decision (or decisions), and the application never retrieves additional data from a database. This encourages good architectural practice since only the application that “owns” the database should access the data directly. But it also maximizes the performance of the Determinations Server by reducing resource contention for database connections, and eliminates the need for both inbound and outbound network connections from the application server.

Scalability

The runtime components of Oracle Policy Automation are available as both pure Java and .NET web applications that take advantage of the standard application scalability approaches of those platforms.

Oracle Web Determinations provides an interactive web application for self-service guidance that uses deployed policy models. Standard scalability techniques are supported, such as clustering servers behind a load-balancing router, or employing an application server's native software load-balancing capabilities. Scaling up on a single server is also very efficient.

Oracle Determinations Server provides a coarse-grained web service that also fully supports standard scalability approaches. Furthermore, because Oracle Determinations Server is completely stateless, no server affinity is required, and every request can be dynamically routed as needed to whichever server is currently experiencing the least load. The processing capabilities of multi-core and multi-processor are also efficiently utilized as available.

Batch Processing Capabilities

Batch processing can be easily achieved in one of two ways. Oracle Determinations Server allows multiple cases to be passed in a single request, and the results for each case are then all handled within that one request. This prevents the need for multiple web service round-trips to Determinations Server, eliminating the latency caused from making multiple calls.

Oracle Determinations Engine Data Source Connector processes in a multi-threaded batch execution mode. It loads processing data from one or more CSV files, and saves the outcomes. Because this approach operates without any web service overhead at all, it can deliver the highest throughput for a high volume batch processing scenario.

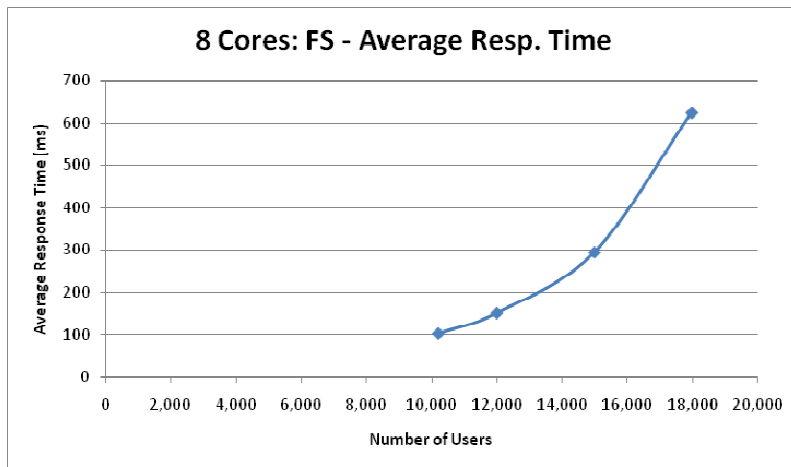
How Fast is Oracle Policy Automation?

The short answer is *fast*.

Below are some results from benchmarking tests conducted using Oracle Policy Automation 10.1. For more details on the testing methodology and policy models used, see the appendix.

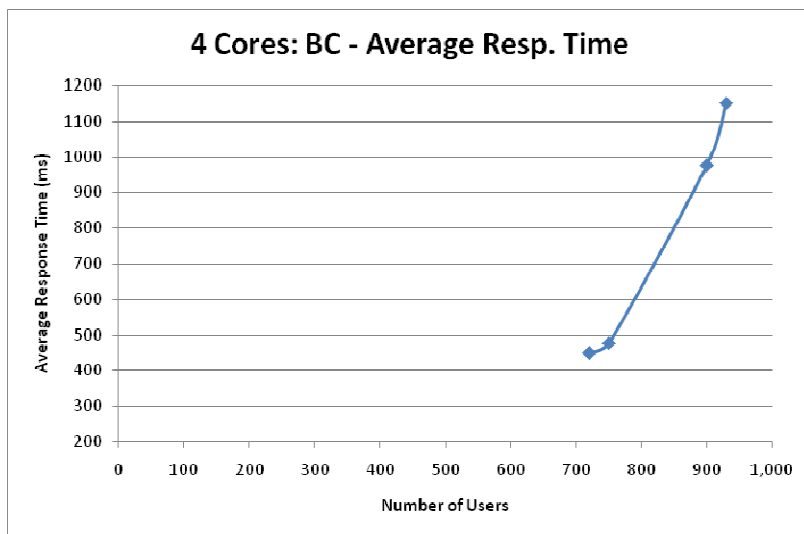
Oracle Determinations Server – Concurrent Users

This section shows the average response time from Determinations Server under varying concurrent user loads. Both tests used a Sun server, with 8 cores enabled of a single SPARC T2+ chip, and 10 second think-time between web service requests for each user.



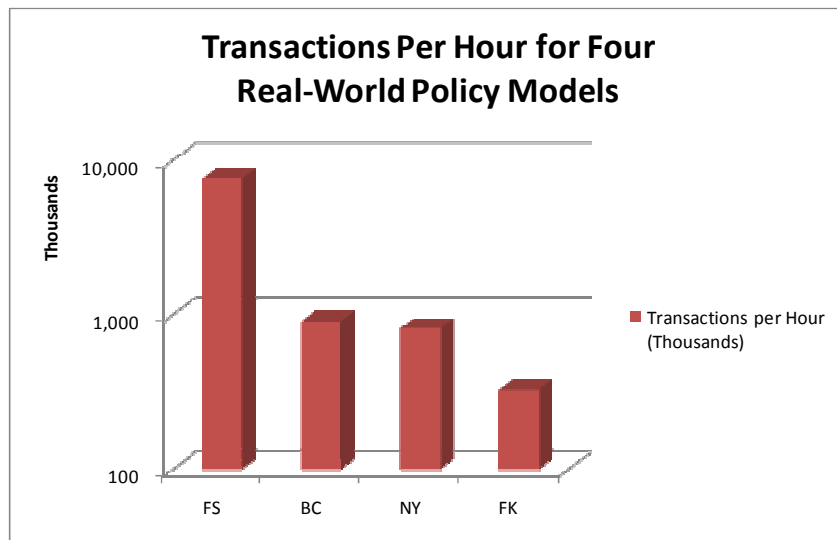
The FS policy model is a simple model with 32 rules. 52 attribute values were used in each calculation. Response time was less than 0.7 seconds at 18,000 concurrent users.

The BC policy model is more complex with 2,172 rules. 106 attribute values were used in the test. Response time was less than a second at 900 concurrent users.



Oracle Determinations Server – Throughput

This section shows the number of transactions per hour processed by Determinations Server, for policy models of varying complexity. Results shown are from tests performed using two IBM servers with 8 dual-core Power6 CPUs enabled each.



As can be seen from this scenario, up to 10s of millions of transactions can be handled per hour using Determinations Server on this hardware. The FS and BC tests were as previously described. The NY tests processed 294 attribute values through 90 rules. The FK tests processed 72 attribute values through 9521 rules.

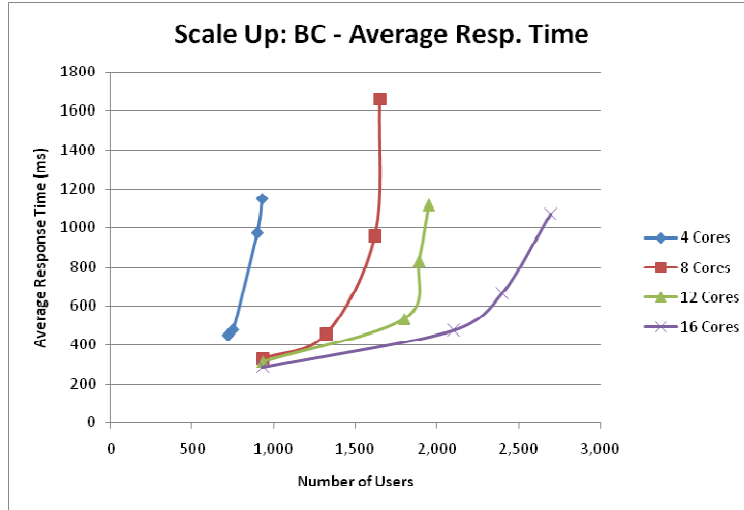
Note that these results were obtained using a *single transaction per web service request*. Bundling multiple transactions into a single request dramatically reduces web service overhead, and improves throughput. No comparative results for bundled web service requests were available at the time of writing this paper, but are planned for a future document.

In general, throughput depends greatly on many factors, particularly the size of the input data and response being processed by the web service, as well as the complexity of the policy model being used. Message size obviously affects the performance of any web service, and Determinations Server is no exception. A detailed analysis of policy model complexity factors is beyond the scope of this paper. Briefly, for Oracle Policy Automation the number of rules generally has much less impact on overall performance than the number of entity instances, the richness of relationships in the input data, and how the relationships between entity instances are used by the policy models. Also, even though *temporal reasoning* is an extremely efficient way of dealing with the complexity of time-sliced calculations, it does have an effect on performance that varies depending on the granularity and number of time periods, and the nature of the rules that interact with those periods.

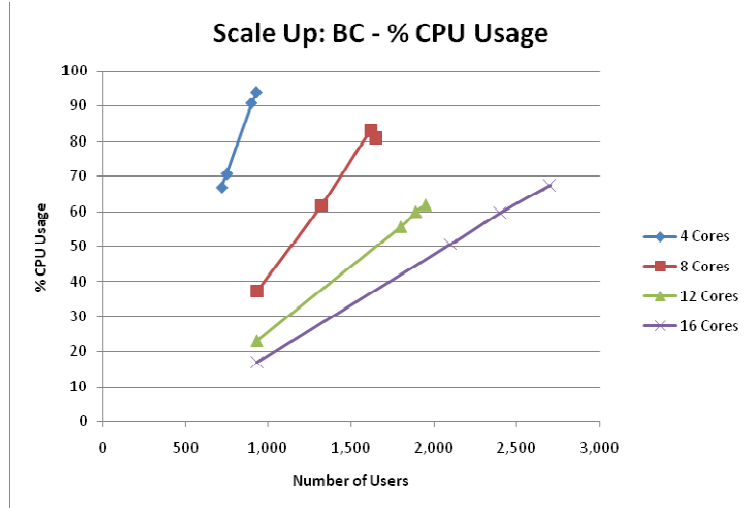
Oracle Determinations Server – Scalability

The following results demonstrate the scalability of Oracle Determinations Server as Processors are added, and as processing load is shared across multiple servers. Tests were performed on a Sun server with two 8 core SPARC T2+ processors.

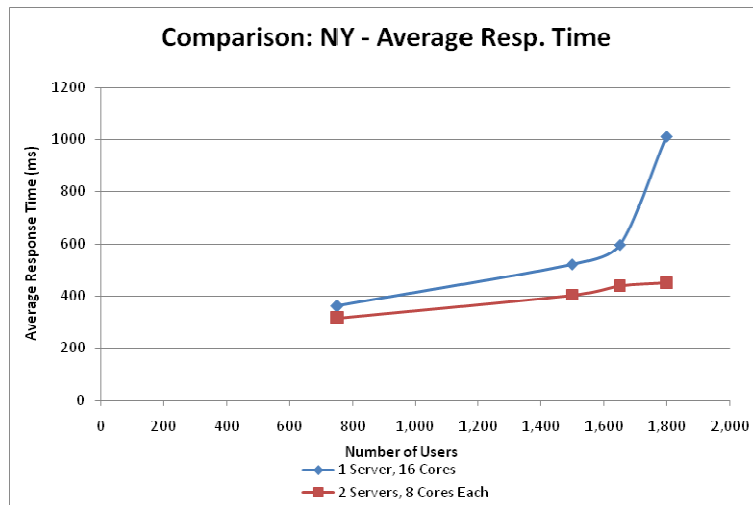
The first test demonstrates that increasing the number of processor/cores available to Determinations Server greatly increases the number of concurrent users that can be supported at a target response time.



This second chart shows that while CPU utilization decreases as the number of processors/cores increases, good use is made of additional processing power as it is made available.



The last chart in this section compares scale-out on 2 Sun servers with one 8-core T2+ processor enabled vs. scale-up on a single server with 2 8-core T2+ processor.



As expected, multi-server scale-out is noticeably more efficient beyond a certain number of users, but up to that point performance is comparable. This again shows that Determinations Server makes good use of multi-processor / multi-core architectures.

In summary, a combination of scale-up and scale-out approaches are recommended to maximize performance and reliability. Scale-out brings the advantage of supporting active-active or active-passive server fail-over.

Oracle Determinations Engine Data Source Connector – Throughput

The Oracle Policy Automation Data Source Connector is designed to maximize batch processing throughput on flat-file data that is provided in CSV files. An example of performance achieved with this technology is as follows:

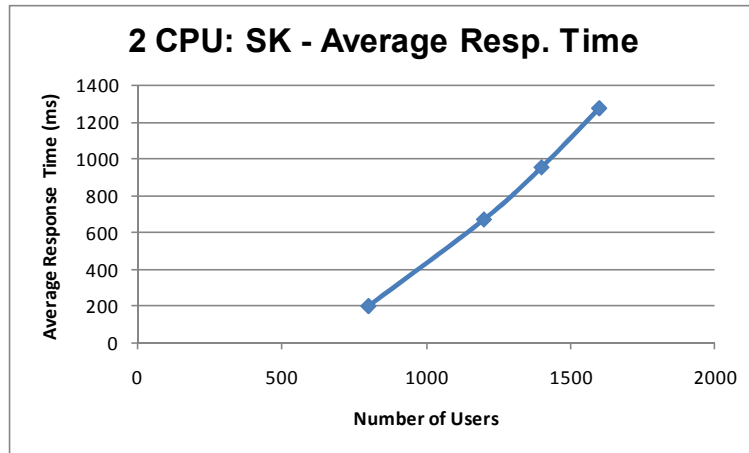
- 5 related CSV files, containing 95 total columns of data
- Approximately 75,000 records total
- Hardware: dual-core Intel Xeon processor (desktop)
- Policy model: 36 rules, doing aggregation and calculation, including custom functions
- Processing time: 43 seconds
- Average CPU utilization: 53%
- Maximum threads: 8

As can be seen from these results, high throughput of millions of transactions per hour can be achieved using this technology.

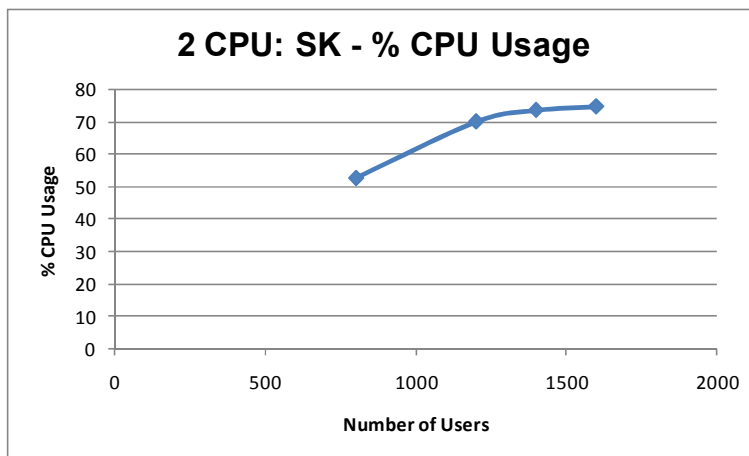
Oracle Web Determinations – Concurrent Users

Oracle Web Determinations makes it simple to deliver scalable self-service interviews that can cope with times of peak demand. This section shows the results of simulating thousands of

concurrent users interactively entering data and reaching a decision on eligibility. The tests were performed on a single IBM server with 2 dual-core POWER6 processors. Users interacted with a multi-screen interactive interview built on the “SK” policy model with 211 rules, and had a 10-second think time between user screens.

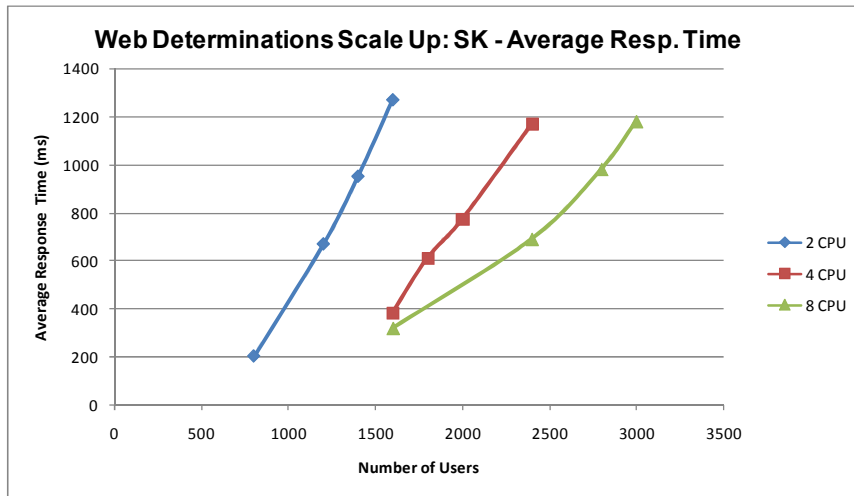


As the response time and CPU usage charts show, nearly 1500 concurrent users can be handled for this application, on this hardware, with sub-second response time and acceptable CPU load on the server.



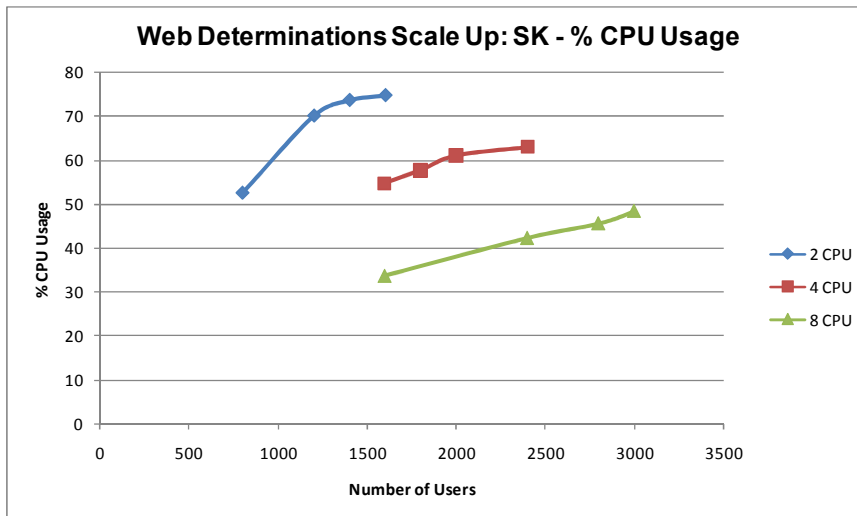
Oracle Web Determinations – Scalability

Web Determinations also scales well as additional processors/cores are made available. Using the same application and test scenario as the previous section, the following chart shows the increase in the number of concurrent users supported, as the number of dual-core processors in use increased.

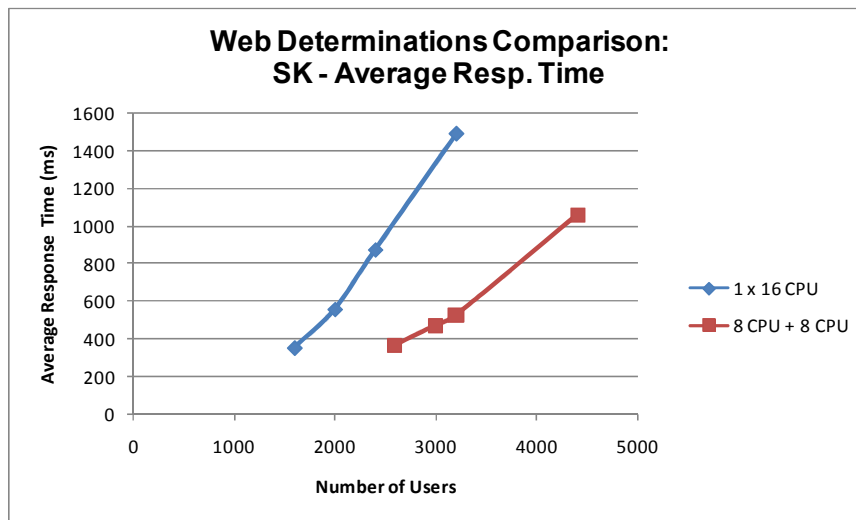


This shows the server handling around 2800 concurrent users at sub-second response time. Think time was again 10 seconds between page requests.

The following chart clearly shows that increasing the number of processors available for handling a self-service application allows more concurrent users to be supported due to lower CPU utilization at the same number of users.



The following chart compares the performance of this same Web Determinations scenario running on a single 16 processor server vs. two 8 processor servers. It is clear that this with this many processors it is worth sharing the load across multiple physical servers, as the two 8 processor servers significantly out-perform the single 16 processor server.



What are some Examples of Real-world Oracle Policy Automation Performance?

HMRC

The United Kingdom's Revenue and Customs agency handles 22 million calls annually from taxpayers that relate to tax credits. Call center staff rely on a guided experience in Oracle Web Determinations to provide accurate advice for each tax credit query. Every day, thousands of call center staff handle many tens of thousands of these customer interactions.

DiaC

Australia's Department of Immigration and Citizenship is responsible for processing 750,000 visa applications annually for visitors or immigrants seeking to come to Australia. Each such application is now processed using Oracle Determinations Server to determine visa eligibility. Also, the public-facing DiaC web-site now offers an interactive Visa Wizard that allows applicants to pre-assess their eligibility for over 70 different visa subclasses, and help them identify the correct visa type to apply for. This solution uses Oracle Determinations Server to deliver this specific advice with a highly tailored user experience. It handles several thousand users every day, by invoking the Oracle Determinations Server web-service.

CBS

The Dutch Bureau of Statistics uses Oracle Policy Automation as a primary modeling solution for statisticians. Policy models are run over tens of millions of records containing data about every household in the Netherlands. By using the Oracle Determinations Engine Data Source Connector, processing to update statistical metrics is now able to complete easily within the target 24 hour window. The largest project uses more than 1100 rules to process 400 pieces of data about 16,000,000 people. By using OPA, processing time has been dropped from 125

hours to 4 hours. CBS also uses Oracle Determinations Server to perform real-time updating of statistics as incoming data is received.

Forsakringskassan

In Sweden, more than 10,000 dentists submit claims on behalf of patients on a daily basis, which are then processed via Oracle Policy Automation Connector for SAP JCo to determine eligibility for reimbursement for dental treatment. When this system went live 70,000 cases were processed in the first 6 hours. Currently, around 20,000 cases are processed per day.

IRS

The United States Internal Revenue Service has implemented advice calculators that cover more than 89 topics, and has made them available to 20,000 internal staff, including their primary call centers. These are used to handle incoming queries on a daily basis. Delivered using Oracle Web Determinations, these calculators are now also available for citizens on the IRS's public facing web-site.

Conclusion

Oracle Policy Automation provides a high performance, scalable and proven solution for your enterprise policy automation needs. It is designed from the ground up to deliver breakthrough performance on complex policy models, and to be easily integrated into your enterprise end-to-end business processes. Providing self-service, web service and batch processing capabilities, Oracle Policy Automation has a solution that will suit even your most demanding business problems. For more information on Oracle Policy Automation, or any of the information presented in this whitepaper, please contact us today.

Appendix

Server Configurations Used for Performance Tests

Sun Servers: Sun SPARC Enterprise T5240, with 2 x UltraSPARC T2+ Processors (8 cores/chip, 8 threads/core) running at 1.6GHz, with 8KB (D) + 16KB (I) L1 cache per core and 4MB L2 cache per processor. One server had 32544MB of system memory, the other 65312MB. Both had a 120GB hard-drive, running Solaris 10 64-bit. Oracle WebLogic Server 11g (10.3) was used as the application server.

IBM Servers: IBM pSeries 5951 partitioned into two identical servers: 16 x 5.0 GHz POWER6 Processors running at 5GHz, with 64K (D) + 64KB (I) L1 cache, 8MB L2 cache and 32MB L3 cache per core(L1) or processor(L2/L3). 65536MB of system memory, 4 x 143GB hard-drives, running IBM AIX 6.1.3.1. IBM Websphere Application Server 7.0 was used as the application server.

Policy Models Used for Performance Tests

Four different policy models were used in the above performance tests. Each policy model had different characteristics as noted. These policy models were based on real world business scenarios of Oracle Policy Automation customers.

FS: Calculates the fraud score of an insurance applicant. Policies were predominately captured using an Excel rule table. Request size was 3.4Kb, response size was 2.7Kb. 34 policy inferences in each request.

NY: Calculates the eligible pension payment for an applicant. All policies were captured in Word rule documents. Request size was 45Kb, response size was 340Kb. 270 policy inferences in each request.

BC: Calculates if a household application is eligible for social assistance. All policies were captured in Word rule documents. Request size was 15Kb, response size was 290Kb. More than 1,200 policy inferences in each request.

FK: Calculates if an applicant is eligible to receive reimbursement for medical treatment. Policies are modeled in a mixture of Excel and Word rule documents. Request size was 12Kb, response size was 13Kb. 74,000 policy inferences in each request.

SK: Calculates eligibility for social assistance. Policies are modeled mostly in Word rule documents, with one Excel rate table. Request size was 30Kb, final decision report was 2Mb. 211 policy inferences in each request.



Maximizing Performance and Scalability of a
Policy Automation Solution
June 2010
Author: Davin Fifield

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

SOFTWARE. HARDWARE. COMPLETE.