SunGard BRASS

Speeding transaction throughput
using powerful Solaris Operating System tools and utilities

Customer Success Story

Industry

- Financial Services

Business Issues

- Today's dynamic marketplace demands trading solutions that can keep pace.
- In financial services environments, applications that experience even a few microseconds of latency can have major implications for trading firms.
- Applications used for trading must be able to handle high volumes and meet performance requirements for transaction throughput.
- Applications designed for massive trading volumes and throughput demand hardware systems capable of providing similarly high performance levels.

Solution

SunGard BRASS is a comprehensive solution for trade order, execution, and compliance management. Running SunGard BRASS on Sun's SPARC® and x64 platforms provides a robust, flexible, and scalable trade processing infrastructure capable of maximizing the powerful software component features.

Business Results

- In recent testing, Sun and SunGard engineers utilized features of the Solaris™ 10 Operating System (OS) to monitor and analyze end-to-end performance of the SunGard BRASS application running on Sun systems.
- Using feedback gleaned from analysis tools, Sun and SunGard engineers tuned the Solaris 10 OS and SunGard BRASS applications and increased transaction throughput to a rate of four times incoming market data feeds.

URL Reference

sun.com/partners
sungard.com/brass

Running the SunGard BRASS application suite on Sun systems provides an integrated platform for trade order, execution, and compliance management needs that leverages the extreme performance, reliability, and scalability of Sun servers and the Solaris™ 10 Operating System.

Success at a glance

With today's financial firms facing exponential increases in market data and regulatory requirements for processing transactions in real time, SunGard was interested in Sun's expertise in tuning applications for large-scale trading deployments. Working together, Sun and SunGard looked for ways to optimize SunGard BRASS applications on Sun platforms in order to help enterprises process more transactions in less time. As a goal, SunGard wanted the BRASS platform to be able to handle processing for between three and four times existing real-time market data feeds.

Testing and tuning methodology

While no two businesses, environments, or even workloads are ever the same, it can be helpful to look at performance in a given environment to understand how the SunGard BRASS applications running on Sun servers behave under certain conditions. Sun and SunGard engineers recently ran tests to determine how the solution can be tuned for improved performance.

Sun and SunGard engineers employed an iterative approach that consisted of using the Solaris Dynamic Tracing (DTrace) software to monitor and analyze existing BRASS systems end-to-end under a typical workload in order to pinpoint possible areas for improvement.

Using DTrace, engineers highlighted different areas of concern, made modifications, and reran tests. At the same time, engineers reviewed system parameters and tuning for the Solaris 10 Operating System (OS) kernel. Once the testing and tuning was complete, the engineers formulated several best practices for analyzing and tuning the Solaris 10 OS and SunGard BRASS applications in order to maximize system throughput.

The testing configuration

All tests used the following systems to stress the SunGard environment.

Sun Blade™ 8000 modular systems

With a choice of server modules that employ single and multicore processors from Sun, AMD™, or Intel®, and the ability to run the Solaris OS, Linux, and Windows operating systems, Sun Blade™ 8000 modular systems offer maximum flexibility. Different server modules can be mixed and matched within a single chassis, enabling organizations to deploy server modules based on the processors and operating system that best serves application requirements and the environment. Modules can be deployed and redeployed as needs dictate.

The memory and I/O subsystems of the server modules offer up to twice the capacity of competing blade and rackmount systems for optimal processing within a smaller datacenter footprint. Depending on type, server modules can scale to 64 GB of memory. With up to 32 lanes of PCI Express bandwidth to deliver to available I/O expansion modules, each supported server module in a Sun Blade 8000 modular system can reach throughput up to 160 gigabits per second.

Sun Blade 8000 modular systems can help to reduce datacenter costs by operating with greater power efficiency and minimized power and cooling needs.

Sun Fire™ V240 and Sun Fire V440 servers

With up to four UltraSPARC® IIIi processors, 32 GB memory, dual 10/100/1000 megabit per second Ethernet ports, and up to four Ultra320 SCSI technology-based disks, these workhorse servers were used for the testing effort. Containing hot-pluggable disks and power supplies, and the Sun Advanced Lights Out Manager, the Sun Fire™ V240 and Sun Fire V440 servers are designed to optimize system uptime, making them ideal datacenter servers.

Analyzing the environment

To analyze the SunGard BRASS application and system environment, engineers used the innovative DTrace facility. DTrace provides a dynamic tracing framework that traces various system attributes in real time on production systems. With a system-level view of application and kernel activities, DTrace runs dynamically without modifying or restarting applications or rebooting the system. Organizations gain the ability to monitor system events and pinpoint areas where applications can be optimized or enhanced.

Testing with other Solaris OS tools and utilities

Along with DTrace, the Solaris 10 OS contains powerful system monitoring tools and utilities that can be used for detailed system analysis. The most commonly used utilities report on memory (vmstat), I/O (iostat), network (netstat), processors (mpstat), and process statistics (prstat).

These critical utilities can help determine the number of disk reads and writes per second, average disk response time, system paging and swapping, system events, interrupts, faults, context switches, thread migrations, system calls, percentage of idle time, and more. By using these tools in conjunction with DTrace, users can compile a comprehensive picture of system performance and uncover vital information concerning areas for improvement.

What the tests revealed

In the SunGard BRASS application, daemon processes receive market data feeds in the form of User Datagram Protocol (UDP) packets from the NASDAQ exchange. The daemons convert the packets to Transmission Control Protocol (TCP) packets and the upstream feed handler processes deliver the converted TCP packets to the downstream processes.

By running an iterative process of testing and application and system modifications, Sun and SunGard engineers pinpointed a number of areas that hampered optimal performance of the BRASS applications. Issues were identified within the application and in several Solaris OS kernel parameters that could benefit from tuning.

Process and memory bottlenecks

The initial DTrace analysis revealed several SunGard BRASS single-threaded processes with bottlenecks. Various BRASS components experienced large queues in spite of processing that occurred at twice the rate of market data throughput.

"In today's volatile markets, it is critical that all of our systems are able to handle exceedingly high transaction volumes and surpass performance requirements with regard to system throughput. We strive to provide our customers with innovative, reliable infrastructure to keep their desks up and running. The Sun project has helped us ensure our customers obtain top performance from all of our systems, spend less time troubleshooting and more time doing what they do best — trading."

Satish Mujumdar
Chief Operating Officer, SunGard

As a result, the initial end-to-end throughput of approximately 600,000 messages per minute could not sustain the kind of high transaction volumes and processing speeds necessary to handle trades in real time. DTrace found the cause — some of the downstream processes used inefficient system calls. Substituting more efficient system calls resulted in throughput of 1,200,000 messages per minute, or twice the previous throughput.

Further testing revealed a bottleneck with the 100 Mb network connection. The Sun and SunGard engineers upgraded the network infrastructure to support Gigabit Ethernet. Even with the Gigabit Ethernet network in place, the systems were unable to run any faster. DTrace testing revealed that the problem was due to upstream feed handler processes spending significant amounts of time acquiring locks for standard memory allocation. To improve performance, Sun engineers recommended replacing the default Solaris OS memory allocator with the more efficient libumem memory allocation library.

Packet transmission

The next area highlighted by testing with DTrace and Solaris OS monitoring utilities showed many UDP packets being dropped during streaming from the NASDAQ exchange. The upstream feed handlers sent packets much faster than the single threaded downstream processes could consume them. As packets were dropped, the system requested retransmission of lost packets from the exchanges, slowing down the system.

In addition, each CPU running the feed handlers experienced involuntary context switches. Higher priority threads pre-empted the feed handlers and prevented completion of jobs within a given CPU time slice.

Sun engineers recommended changing the CPU scheduling class for the feed handlers to the fixed-priority and increasing the feed handler processes' priority to 60. In addition, the UDP and TCP buffer sizes and highwater marks were increased in the Solaris 10 OS kernel to help prevent downstream packet drops.

DTrace also showed TCP throughput latency caused by Solaris 10 OS Internet protocol (IP) code trying to automatically adjust the interrupt rate due to multiple UDP applications accessing the same Gigabit Ethernet interface. Engineers disabled the Solaris 10 OS kernel parameter tx_interrupt_enable to help improve the transmit performance of applications utilizing the UDP protocol.

CPU utilization

As a result of modifications made during testing, throughput increased to three and a half times the rate of incoming market data. SunGard engineers subsequently eliminated one process requirement, enabling the feed handlers to run at over four times the rate of incoming data. Beyond that point, DTrace and the Solaris OS monitoring utilities showed that the single-threaded downstream processes were running out of available CPU resources.

To solve the problem, Sun engineers created processor sets on the systems running the downstream processes. Each downstream process was bound to a set, providing dedicated CPU resources and reducing the context switch-related latency associated with changing CPUs.

Interrupts were disabled on the processor sets, the CPU scheduling class was set to fixed-priority, and the downstream processes were given highest priority in order for the processes to receive fixed CPU slices and avoid process interruption.

The daemons receiving UDP packets from the NASDAQ exchange and performing conversions to TCP also suffered from the same issues as the feed handler processes. Sun engineers recommended the same solutions of changing the CPU scheduling class to fixed-priority, increasing the daemons' process priority to 60, and increasing the UDP and TCP buffers and highwater marks to resolve the daemons' performance issues.

Engineers using DTrace also discovered that system monitoring tools used by the SunGard BRASS application read from the system /proc folder when taking performance snapshots and risked stopping the application while it was running.

Best practices

Armed with the results of the testing effort, Sun and SunGard engineers formulated several recommendations for best practices in application development and system tuning.

- Take advantage of multithreading capabilities by modifying single-threaded application processes where possible.
- Handle memory allocation for multithreaded processes using the fast, scalable, object-caching libumem library which incorporates multithreaded application support.
- Create processor sets and bind processes to the newly created sets when dedicated CPU resources are required. Processes that do not need dedicated CPU resources, but that require adequate CPU time slices, benefit from being assigned a fixed-priority CPU scheduler class with the highest possible user global priority. Doing so can help improve the transmit performance of applications utilizing TCP connections.
- Set tx_interrupt_enable to zero in the Solaris 10 OS if there are multiple UDP applications accessing a single Gigabit Ethernet interface.
- Use Gigabit Ethernet when accessing financial services market data in order to speed transmission rates.
- Tune the Solaris 10 OS UDP and TCP kernel parameters to help increase network throughput and reduce latency. Increasing the UDP and TCP buffer sizes and highwater marks can help to eliminate dropped packets.
- Develop applications with the Sun™ Studio 12 compiler software to leverage processor-specific optimizations, including processors from AMD and Intel. Using the -target, -xchip, and -xarch flags enables applications to take advantage of chipset optimization.

Conclusion

Application developers can take advantage of powerful tools and utilities within the Solaris 10 OS to monitor and analyze system and application performance, pinpoint problem areas, and help to correct performance issues. Innovative features such as DTrace help administrators to observe applications dynamically, and gain insight into application and operating system kernel activities at a level previously unthinkable. The result is reduced time, effort, and expense for application troubleshooting and enhancement, culminating in optimal system and application performance.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN **Web** sun.com