TECHNICAL DOCUMENT

ALTAIR® GRID ENGINE® BENCHMARK TECHNICAL BRIEF

Distributed resource management (DRM) systems are used extensively in many different industries including pharmaceuticals, oil and gas, automotive simulation, and electronic design automation (EDA). DRM is critical to business competitiveness. Altair® Grid Engine® is a DRM solution that enables efficient product design for measurable economic benefits such as faster time to market, fewer defects and warranty returns, and safer and more reliable products.

Executive Summary

Altair continually optimizes throughput and performance of applications, containers, and services across all types of shared resources, both on-premises and in the cloud. With that goal in mind we benchmarked Altair Grid Engine against previous versions of Grid Engine to ensure top performance and efficiency.

Performance and throughput optimization boosts efficiency

Altair Grid Engine significantly reduces overhead

In this brief we compare Altair Grid Engine 8.5.0 to Grid Engine 6.2U5, the last version from Sun Microsystems. The testing results show that at a minimum Univa Grid Engine 8.5.0 is two times (2x) faster than Grid Engine 6.2U5 on many of the tests we performed. In the most difficult test Altair Grid Engine 8.5.0 is 9.5 times faster than Grid Engine 6.2U5. This means that under normal conditions overhead is significantly reduced, resulting in measurable differences in system responsiveness workload completion time. In summary, Altair Grid Engine exhibits has dramatically improved system performance and scalability over Grid Engine 6.2U5. Altair continues to improve Altair Grid Engine delivering a faster, more scalable and responsive soution.

Introduction

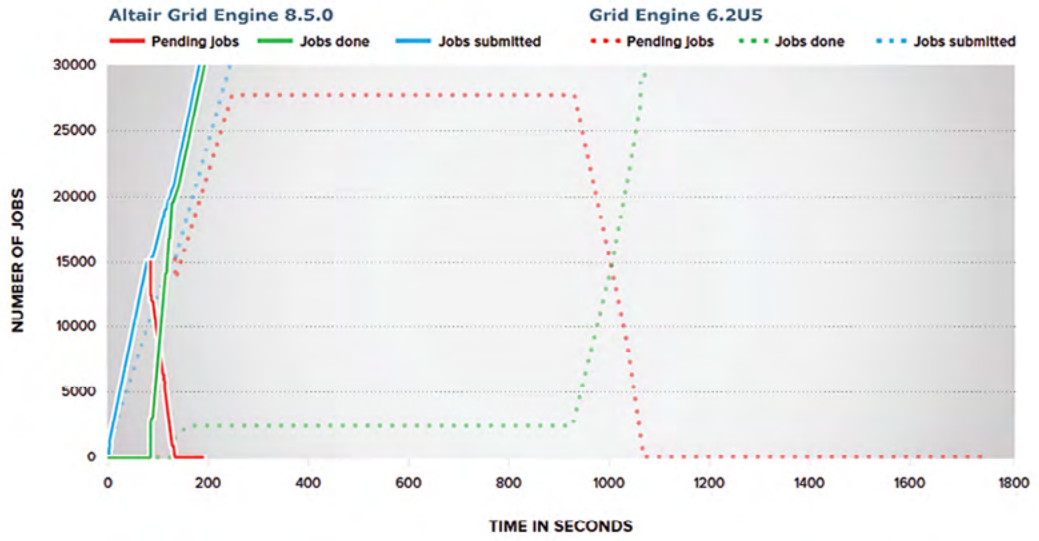
This brief will show the methodology and results of testing Altair Grid Engine 8.5.0 against Grid Engine 6.2U5. Grid Engine 6.2U5 forms the basis of all Grid Engine variants and has remained unchanged for many years with no major improvements. Altair acquired Grid Engine engineers, technology, customers, and assets and now represents the successor to Sun Grid Engine, Oracle Grid Engine, and Univa Grid Engine. The original Grid Engine team continues to develop the software and add new features. As part of our development and release process we continually test and compare new releases of Altair Grid Engine to ensure that performance, scalability, and responsiveness of the system continues to improve with each new version.

Key Results

For brevity, only key and more complex test results are included. For the complete results, please visit www.altair.com/contact-us.

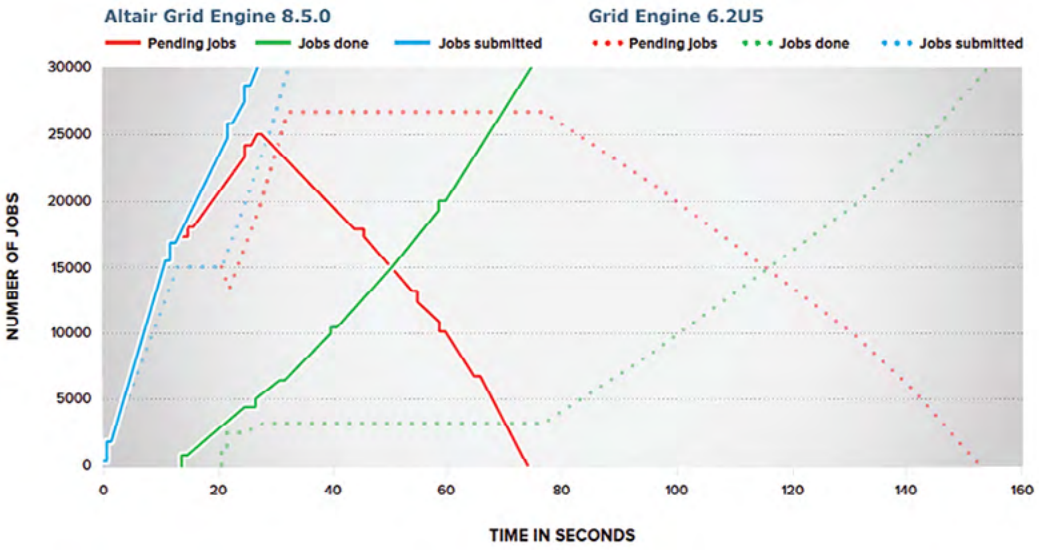
1. Mixed job submission with many clients

Altair Grid Engine is 9.5x faster at completing the same set of work. (Note: It is clear that with many clients Grid Engine 6.2U5 begins to slow dramatically.) The graph below clearly shows that the number of clients that query the system affect Grid Engine 6.2U5. The system stalls for a considerable amount of time processing few jobs, then suddenly at 1,000 seconds it begins to process again. Altair Grid Engine displays an almost perfect response line. As soon as the queues are opened at 15K jobs it immediately starts running and finishing jobs.



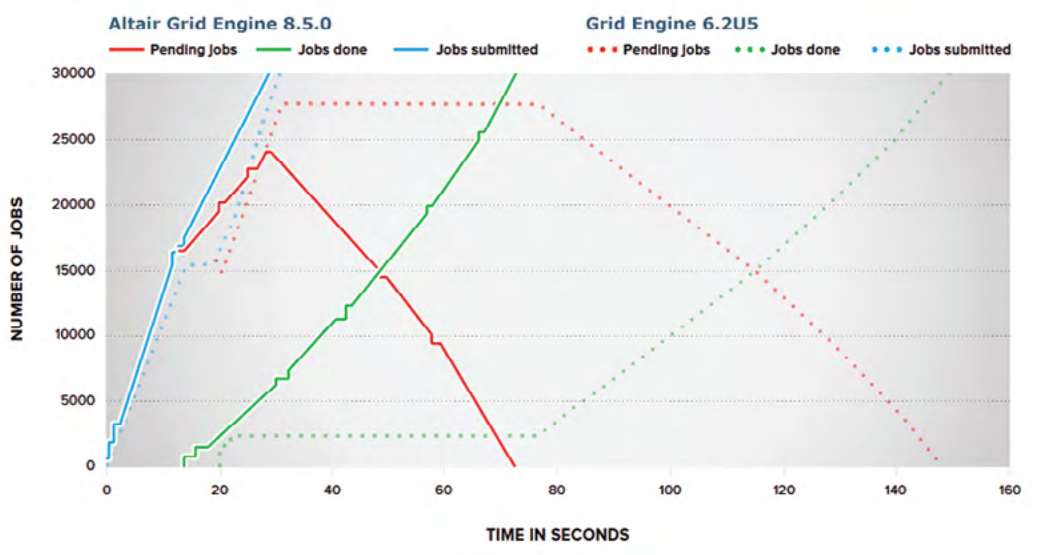
2. Mixed job submission with Share Tree algorithm

Altair Grid Engine 8.5.0 is 2x faster than Grid Engine 6.2U5 at completing the same set of work. In the graphs below both versions of Grid Engine exhibit similar graphs for submitted jobs but differences appear when you compare the jobs pending and jobs done and finally Altair Grid Engine finishes much earlier.



3. Mixed job submission with reservations

Altair Grid Engine is 2x faster than Grid Engine 6.2U5 on the same workload. Resource reservations are particularly time consuming in all versions of Grid Engine, however in Altair Grid Engine you can see from the graph below that once 15K jobs are submitted the system begins processing, but more slowly than without reservations. Grid Engine 6.2U5 'stalls' for approximately a period of 40 seconds doing minimal work.



Methodology

The benchmark process begins with these simple tests and evolves to more complex and demanding tests. The tests are designed to simulate new clusters running jobs, existing clusters receiving many new jobs, clusters with simple policy configuration and clusters with complex policy configuration. Finally, the most demanding test combines many of the above while simultaneously simulating thousands of client commands (qstat, qhost, qconf) continuously calling the system. This last test simulates a cluster that is processing large volumes of work while serving many user requests. This type of scenario typically happens in large Grid Engine deployments. All tests measure the overhead added by Grid Engine and minimize the overhead caused by the work submitted to the cluster. This is accomplished by submitting jobs that sleep for zero seconds and exit.

Test environment – All tests are performed on Amazon EC2 and can be replicated quickly to test different versions of Grid Engine. Clusters are not reused. A new cluster is created for each test run.

Cluster Location	Amazon EC2
Number of Nodes	200
Type of Master Node	4 vcpu, 30.5Gb, r3.xlarge, 80Gb SSD
Type of Compute Nodes	1 vcpu, 3.75Gb, m3.medium, 4Gb SSD
Network Configuration	Master: 800 Mbps, Compute: 391 Mbps
Network Filesystem	NFS for home directories
Master Spooling	Local: BerkeleyDB
Compute Spooling	Local to each compute node
Queue Configuration	1 queue, 4 slots each host = 800 slots
Type of Job Submitted	Sleeper job with zero seconds 'sleep 0'
Operating System	Master: CentOS 6.8 Final, Compute: CentOS 6.7 Final

Test submission types

1. **DISABLED**: All queues are disabled. 0K jobs are submitted into the system from 50% of the submission hosts, no jobs are submitted from the master. Once all jobs are submitted the queues are opened and all jobs are processed.
2. **ENABLED**: All queues are enabled. 30K jobs are submitted into the system from 50% of the submission hosts, no jobs are submitted from the master. The system immediately begins processing jobs as they arrive.
3. **MIXED**: All queues are disabled. When 15K jobs are submitted into the system from 50% of the submission hosts (50%) the queues are enabled. The system begins to process jobs and also accept new jobs until 15K more jobs are submitted and total of 30K jobs are processed.
4. **ARRAY**: All queues are disabled. Jobs are submitted as arrays of 500 jobs each. When 30 array jobs are submitted the queues are enabled. The system begins to process array jobs and accepts new array jobs until 60 array jobs of 500 tasks each are processed.

Test policy types

1. **FUNCTIONAL**: Grid Engine functional policy is turned on for test
2. **SHARETREE**: Grid Engine hierarchical fairshare policy
3. **MANY_CLIENTS**: Many clients each submitting queries to the system in a continuous loop while the system operates
4. **RESERVATION**: Resource reservation turned on in scheduler with a maximum of 160 reservations

In our testing we combine the above test submission types and test policy types to ensure better coverage in our testing and to increase the complexity of testing from the very simple test case to the most complex high throughput with lots of clients test case.

Conclusions

From the tests comparing Altair Grid Engine 8.5.0 to Grid Engine 6.2U5 we can clearly see that Altair Grid Engine outperforms Grid Engine 6.2U5. Altair Grid Engine is 2x faster than Grid Engine 6.2U5. This means that the overhead added byAltair Grid Engine is 50% less than Grid Engine 6.2u5. That wasted time can now be used to run more jobs, or deliver solutions faster. In addition to the 2x performance improvement, Altair Grid Engine is more responsive under heavy loads than Grid Engine 6.2U5. We have anecdotally seen this in the field but the above graphs show it clearly that when the cluster is busy processing work and end users are continuing to submit more work and query the system, Altair Grid Engine responds promptly and continues to process work, whereas Grid Engine 6.2U5 stalls. Under heavy load Grid Engine 6.2U5 is 9.5x slower than Altair Grid Engine at processing the same workload. This is critical for our customers — Altair Grid Engine continues to process jobs efficiently while always responding to incoming user requests.

For more information about the benchmark and results, please visit altair.com/contact-us.