ScaleOut Software

GRID COMPUTING EDITION

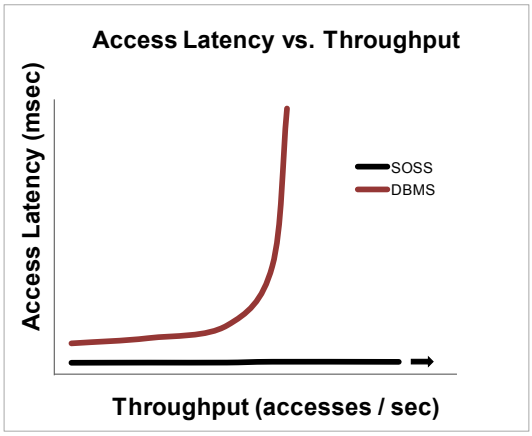
Parallel Data Analysis for Fast Time-to-Insight

Overview

ScaleOut StateServer Grid Computing Edition™ (GCE) combines scalable, memory-based, distributed data storage with powerful computational features to deliver a distributed data grid for server farms and high performance computing environments. Advanced capabilities for rapidly searching grid data and quickly developing scalable grid applications make it perfect for data parallel computation. Also included is the ScaleOut Management Pack™ which provides comprehensive tools for observing, managing, and preserving grid data.

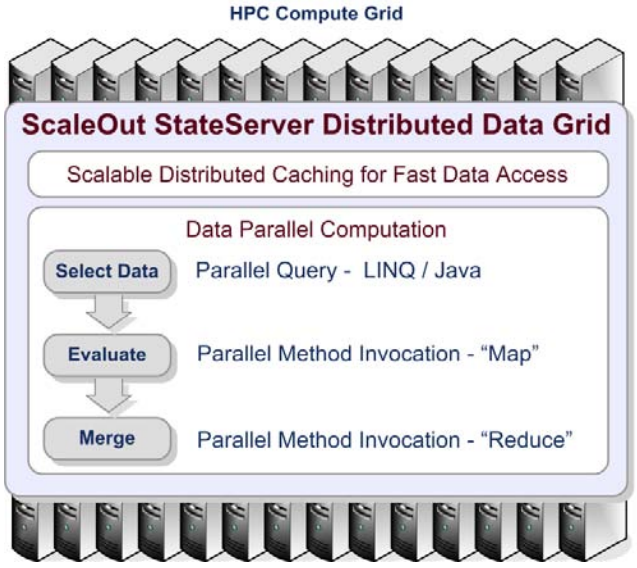
Fast Data Access

Financial services and other data-intensive industries routinely demand real time processing and overnight batch analysis of large data sets. By storing fast-changing data in-memory in ScaleOut StateServer's distributed data grid using industry-leading distributed caching technology, these applications can dramatically reduce access latencies, eliminate access bottlenecks, and achieve peak performance for your grid computing applications. As illustrated by the graph to the right, the grid's throughput can be scaled by adding servers as needed to handle the largest applications while maintaining low access latency as the throughput increases. The distributed data grid also lets applications immediately share data across compute nodes without the need for message passing; this simplifies program structure and shortens design cycles. These combined benefits make GCE a powerful data access platform for a wide range of high performance computing (HPC) applications.



Powerful Parallel Search

Applications can perform parallel queries to rapidly search grid-based data for selected objects based on properties of stored objects or on user-supplied metadata. C# applications can specify queries using Microsoft's powerful Language Integrated Query (LINQ), and Java applications can make use of composable filter methods, providing powerful and familiar tools for developers. In addition, parallel query of object metadata added by users is supported. Employing patent-pending, parallel search and merging algorithms, GCE provides fast, scalable, and highly available parallel query across all hosts within the distributed data grid.



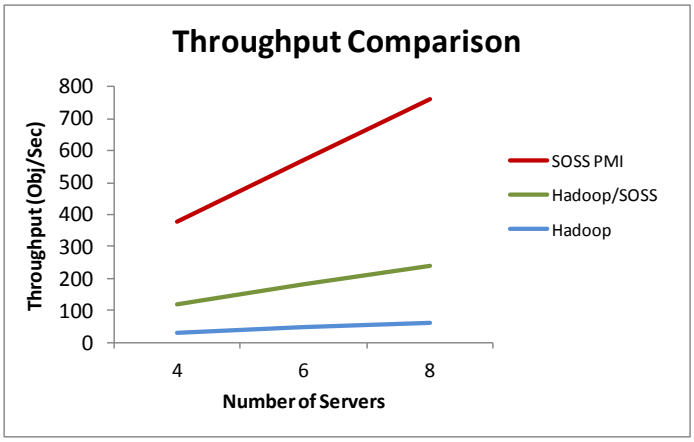
Parallel Data Analysis

ScaleOut StateServer GCE enables scalable, grid-based computation on memory-based datasets to deliver fast insights and eliminate bottlenecks in analyzing large datasets. Applications can perform powerful, "map/reduce" computations by automatically executing user-defined

methods in parallel on a selected set of objects and then combining the results using user-defined merge algorithms. Called "parallel method invocation," this feature provides a compelling implementation of the "map/reduce" programming pattern which simplifies application design and minimizes both data motion and scheduling overhead.

Unlike traditional map/reduce implementations which process file-based data sets, GCE performs map/reduce analysis on memory-based data stored in its scalable, distributed data grid, thus eliminating significant overhead due to file I/O. GCE automatically schedules the user's "map" operation across multiple cores within each server and all servers within the grid. GCE's execution engine maximizes parallelism and minimizes data motion within the grid. In addition, GCE performs automatic parallel merging of the user's "reduce" operation across all grid nodes. Tests have shown that ScaleOut StateServer GCE can deliver significantly higher performance for map/reduce analysis of datasets that are stored in its in-memory, distributed data grid.

For example, GCE demonstrated 16X faster throughput over the open source Hadoop platform for back testing stock trading strategies in a financial analysis application As shown in the graph to the right, SOSS's parallel method invocation (the red line in the graph) eliminated file I/O required by the equivalent Hadoop application (the blue line) as well as batch scheduling overhead. Storing the dataset within a distributed data grid (the green line) improves Hadoop's throughput by up to 6X, but SOSS PMI retains a significant performance advantage by eliminating key overheads.



Importantly, GCE delivers its high performance without the need for the user to install and tune a complex map/reduce infrastructure. Applications can be written as straightforward, in-memory C# or Java methods that avoid explicit grid accesses. Because GCE avoids the use of a batch scheduler, applications can perform map/reduce computations with in-line API calls for near real-time responsiveness. Datasets requiring fast-turn analysis are a perfect match for GCE. When compared to other approaches like Hadoop, GCE's ease-of-use significantly reduces development time while delivering both low latency and high throughput.

ScaleOut StateServer GCE offers industry-leading map/reduce analysis for datasets which fit in memory within its grid servers. Many useful datasets can be held in a distributed data grid. For example, even a small grid of eight servers, each with 64GB of memory, can host in-memory datasets of several hundred gigabytes. With today's low memory prices and the proliferation of cloud computing, very large grids have become economical for hosting and analyzing large datasets. For example, a 500-server grid with 64GB of memory per server can host about 32TB of memory-based storage.

Powerful Management of Grid Data

Included with GCE, the ScaleOut Management Pack adds important functions that extend your ability to manage, analyze, and protect data stored in the distributed data grid. The Management Pack includes two components: an object browser for visually browsing and managing objects stored in the distributed data grid and a parallel backup and restore feature for quickly archiving its contents in the file system. The object browser's ability to visualize the grid's contents dramatically simplifies application development. The parallel backup and restore feature adds the capability to save the contents of a distributed data grid as a snapshot which later can be restored for analysis.