ScaleOut Software

ScaleOut hServer

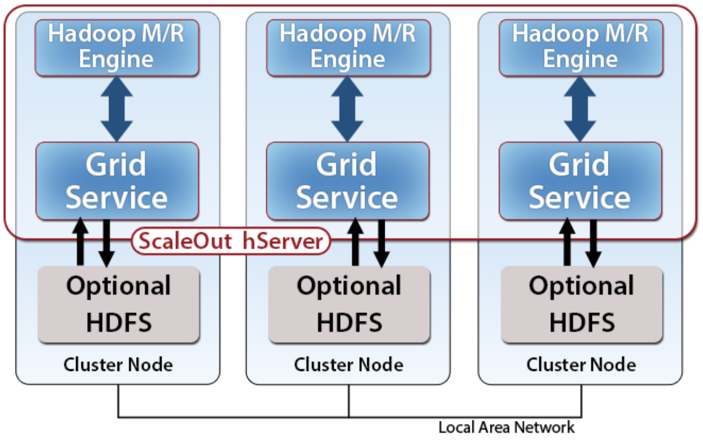
# *The World's First In-Memory Execution Engine for Hadoop*

## Overview

ScaleOut hServer™ is the world's first in-memory execution engine for Hadoop MapReduce. Now you can analyze live data using standard Hadoop MapReduce code, in memory and in parallel – all without the need to install and manage the Hadoop stack of software. Gone are disk I/O latencies, slow start-up times, and software environment management headaches. Benchmark tests have demonstrated **20x faster execution** over the Apache Hadoop distribution. This breakthrough performance enables Hadoop MapReduce to be used in real-time scenarios in financial services, e-commerce, logistics, or wherever results are needed in seconds instead of minutes or hours.

## In-Memory Hadoop for Live Data

Instead of storing "live" data on disk within HDFS, ScaleOut hServer uses a fast, scalable in-memory data grid (IMDG) that enables data to be continuously updated and analyzed using ScaleOut hServer's new Hadoop MapReduce engine. ScaleOut hServer's IMDG middleware stores key/value pairs across an elastic set of networked servers, ensuring fast data access, linear scalability, and high availability. ScaleOut hServer's integrated MapReduce engine executes standard Hadoop MapReduce programs directly in the IMDG, delivering results in seconds so you can spot important trends in your data as they occur. At the same time, your live application can easily create, read, update and delete fast-changing data in the IMDG with easy-to-use Java APIs. Together, these capabilities enable you to bring the power of Hadoop's analytics to live, operational systems.

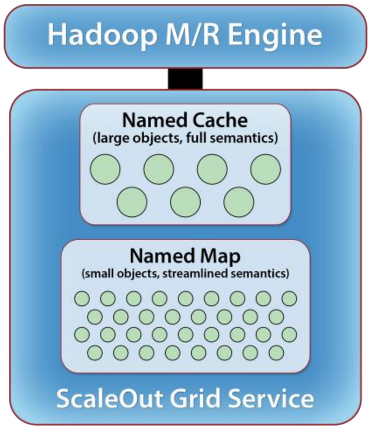


Consider these advantages of using ScaleOut hServer for real-time analysis:

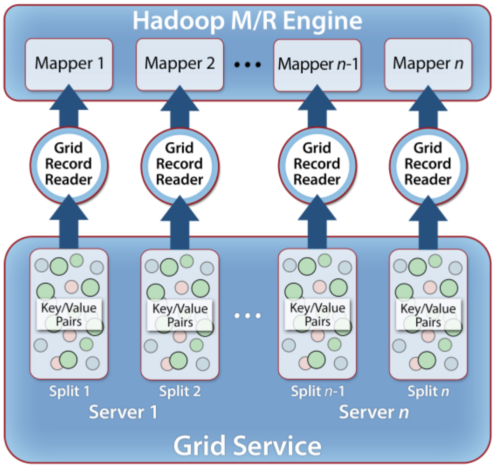
- The Hadoop MapReduce engine executes standard Hadoop code 20X faster in benchmark tests.
- ScaleOut hServer eliminates Hadoop's batch scheduling overhead, resulting in sub-second start-up times.
- ScaleOut hServer's IMDG stores data at in-memory speed, reducing data access times, and is designed to hold fast-changing operational data.
- Optional sorting and optimized combining and data shuffling between the mappers and reducers using in-memory storage streamlines processing and minimizes execution time.
- For memory-based data sets, automatic setting of key MapReduce parameters, such as splits, partitions, and slots, simplifies development and makes MapReduce execution self-tuning.
- Performance linearly scales just by adding servers to increase memory capacity and throughput; ScaleOut hServer automatically rebalances the workload.
- The IMDG ensures that stored data is highly available to protect from server or network failures.
- The IMDG's key/value storage and associated Java APIs match the object-oriented architecture of your Hadoop application. Optimized data storage for large data sets with very small key/value pairs ensures efficient memory usage and maximum MapReduce performance.
- ScaleOut hServer automatically detects and optimizes applications that produce a single, combined result instead of a key/value space. This is particularly useful in real-time applications.
- To enable analysis of large data sets, data can be streamed from HDFS into ScaleOut hServer and results written back to HDFS.
- ScaleOut hServer can serve as a distributed cache for data read-in from HDFS where the data set fits in memory and faster subsequent runs using the same data are needed.

## Optimized, In-Memory Data Grid

After you install ScaleOut hServer, it will automatically discover and self-aggregate into an in-memory data grid spanning a cluster of servers. Using ScaleOut hServer's Java APIs, your application can create, read, update, and delete key/value pairs in the IMDG to manage fast-changing data within your live application, keeping the data in the grid up to date as changes occur in your application.


Hadoop M/R Engine
Named Cache
(large objects, full semantics)
Named Map
(small objects, streamlined semantics)
ScaleOut Grid Service

IMDGs traditionally host complex objects with rich semantics in the grid. However, Hadoop jobs often require storing and analyzing huge numbers of very small objects, such as sensor data or tweet streams. To handle these divergent requirements, ScaleOut hServer supports two object storage mechanisms in its IMDG. Designed for large, complex objects, the *Named Cache* supports rich functionality such as property-oriented query, dependencies, timeouts, optimistic and pessimistic locking, transparent access from remote data grids and backing stores, and much more. With the new *Named Map*, ScaleOut hServer adds Java ConcurrentMap semantics to efficiently organize large populations of small data objects and minimize the amount of metadata associated with each. Objects stored in a named map can be queried in parallel and can be cached in the client using user-adjustable coherency policies. For fast loading and updating of key/value data, the Named Map provides bulk insert and bulk update functions.



In both named caches and named maps, applications create, read, update, and delete individual objects to manage live data. The Hadoop developer now has the choice to store and analyze either heavyweight objects with rich semantics or lightweight objects depending on the type of data being analyzed.

ScaleOut hServer includes a *Grid Record Reader* to input key/value pairs to Hadoop's mappers with minimum latency. Its input format automatically creates splits of the specified input key/value collection to avoid network overhead when retrieving key/value pairs on all worker nodes. The Grid Record Reader works with both named caches and named maps. Likewise, a *Grid Record Writer* enables pipelined output of results from Hadoop's reducers back to a named cache or named map in the grid.

## Beyond Live Data Analysis

In addition to real-time analytics, ScaleOut hServer can be used for fast analysis of large, static data sets, even those that don't fit in memory. Its MapReduce engine efficiently reads and processes HDFS data. In addition, it can be configured to transparently cache HDFS data in the IMDG to reduce access latency in subsequent runs.

In yet another usage model, ScaleOut hServer enables fast, easy "what-if" analyses of static data held in the grid. Its fast execution time streamlines processing in applications, such as financial modeling, which require running repeated MapReduce runs on the same data set. For example, ScaleOut hServer enables stock trading strategies to be easily tested and honed with multiple simulations across price histories held in-memory.

Finally, ScaleOut hServer provides a simple, easy-to-use debugging environment for developing MapReduce applications. After installing ScaleOut hServer in minutes, you can load a subset of your data into memory and execute Hadoop jobs in seconds. This means that you can rapidly iterate on your MapReduce code until you are getting the results you want.

## Community Edition

ScaleOut hServer is available in both a free Community Edition and several commercial editions. The Community Edition is licensed for either evaluation or production use and supports up to a four-server IMDG with a maximum data set size of 256GB. Support for the Community Edition is provided via the ScaleOut Community Forum, where you can ask questions and exchange ideas with other users and ScaleOut Software’s experts. Download the community edition at www.scaleoutsoftware.com/hserver/.