McObject’s New eXtremeDB Cluster Provides a Powerful Distributed Database Solution for Real-Time Applications

McObject announced the release of eXtremeDB Cluster, the first clustering database system built from the ground up for distributed embedded software as well as real-time enterprise applications. eXtremeDB Cluster manages data stores across multiple hardware nodes, dramatically increasing the net processing power available for data management, reducing system expansion costs, and delivering a more scalable and reliable database solution for increasingly data-intensive real-time applications.

In McObject's benchmarks, eXtremeDB Cluster delivered breakthrough performance, including an astonishing 161% throughput improvement when scaling to four nodes from one node. McObject CEO Steve Graves predicted rapid adoption of the new clustering solution, based on input from customers in market segments including telecom/networking equipment, financial markets and Web services/hosted solutions where the company’s core eXtremeDB In-Memory Database System (IMDS) technology is already widely used.

“The new Cluster edition builds on our existing eXtremeDB product family to deliver the most advanced database solution for carrier-grade telecom gear, algorithmic trading applications, software-as-a-service (SaaS) platforms and other applications requiring a high level of speed, scalability and reliability,” Graves said. “eXtremeDB Cluster adds an important new capability by enabling two or more servers to share the workload, eliminating any ceiling imposed by being CPU-bound on a single server.”

With its streamlined in-process (rather than client/server) database architecture, eXtremeDB Cluster eliminates complexity when compared to competing clustering solutions that are offered by relational database management system (RDBMS) vendors, Graves said. eXtremeDB Cluster’s easier set-up cuts out the extensive consulting engagements that are often part-and-parcel of these vendors’ clustering solutions, giving McObject’s product an important cost-of-ownership advantage.

In an eXtremeDB Cluster deployment, every database instance serves as a “master.” Any process on any node can update its local database, and the eXtremeDB Cluster software will efficiently replicate the changes to other nodes in the cluster. This eliminates the potential bottleneck when all changes must be written to a single master (the common solution in High Availability systems), and the increased processing power results in much faster database activity.

The hardware for each node can be a low-cost (i.e. “commodity”) server, so that the system can expand cost-effectively. Distributing the system across multiple hosts ensures continuous availability in the event of a failure on one node. eXtremeDB

Cluster’s “shared nothing” architecture eliminates reliance on a shared SAN or other storage resource. eXtremeDB Cluster supports the same ACID transactions offered by the non-clustering eXtremeDB editions, making it an attractive choice for applications that demand integrity of distributed data.

eXtremeDB Cluster is compatible with eXtremeDB's proven In-Memory Database System, 64-bit, and Fusion (hybrid in-memory and on-disk data storage) editions. It also exploits eXtremeDB’s Multi-Version Concurrency Control (MVCC) transaction management, which eliminates database locking and thus dramatically enhances performance and scalability for multi-threaded applications running on multi-core hardware.

Designed as the first embedded database technology specifically targeting embedded applications, eXtremeDB has enjoyed broad adoption across all embedded software market segments and can be found in millions of deployed devices and systems as well as in non-embedded applications that demand a highly flexible real-time database management system. eXtremeDB’s core in-memory architecture overcomes the I/O and caching bottlenecks inherent in disk-based DBMSs. Its 64-bit support allows in-memory databases to grow to terabyte-plus sizes, while with eXtremeDB Fusion, on-disk database sizes are limited only by available file system space (in either 32-bit or 64-bit implementations). MVCC allows threads to simultaneously execute insert/update/delete operations with the database without having to be queued by a lock arbiter.

For more information please visit, www.mcobject.com [1]

Links:
[1] http://www.mcobject.com