



Progress® ObjectStore® Enterprise is a database management system that delivers extraordinary scalability and responsiveness for mission-critical applications.

FEATURES AT A GLANCE

- > Patented Cache-Forward Architecture that enables highly responsive access to locally cached data.
- > Distributed data caching with complete transaction coordination among the local caches.
- > In-memory data management that leverages the physical hardware in a model patterned after the operating system.
- > Extraordinary read-only data access via non-blocking MVCC mechanism that permits updates simultaneously with data reads.
- > Middle tier class libraries for distributed application server deployments.
- > Native support for both C++ and Java object models.
- > Support for standards-based JDO programming interface.
- > A system administration tool allows users to monitor, control, and report on executing ObjectStore-related processes.

DATA SHEET

Progress® ObjectStore® Enterprise is a highly scalable real-time data management solution that leverages its native support for Java™ and C++ objects—together with a patented, distributed caching architecture—to deliver the speed, reliability, and scalability required for today's mission critical applications in financial services, telecommunications, retail, logistics, government, and e-commerce. ObjectStore has proven to be the ideal database for many applications in high volume transactionintensive environments that demand extremely fast response times. ObjectStore likewise excels in situations that involve complex, application-specific data models such as those found in GIS, CAD/CAM, Life Sciences, Energy Exploration/Distribution, and others.

PATENTED CACHING ARCHITECTURE

The foundation to ObjectStore's capabilities is its patented Cache-Forward™ Architecture (CFA). The ObjectStore CFA has been designed to maximize application performance through data load balancing, cache affinity, transaction services, and component coordination. CFA creates local caches from the ObjectStore server that allow large datasets to be situated close to the application, reducing the network and data access latencies that adversely impact applications operating against other databases.

In addition to the physical proximity of local caching, ObjectStore also accelerates performance by managing the local caches across physical memory and local disk using a virtual memory management (VMM) model patterned after the virtual memory features of the operating system. VMM enables data to be treated as if it were in memory, with responsiveness that corresponds to in-memory speeds. ObjectStore's unique ability to distribute and maintain component data caches, together within its in-memory execution model, enables large-scale systems to outperform traditional server-centric database systems, while retaining the integrity of the data and transactions that are executing in the distributed caches.

To deliver higher distributed performance in a multi-tier architecture, while retaining transactional integrity, CFA includes a callback locking mechanism that caches lock information with the data. This enables many read and update requests that typically require lock coordination with the database server to be handled entirely with data from the cache, without need to access the server and its inherent latencies.



TRANSPARENT PERSISTENCE

Within its support for C++ and Java object models in the database, ObjectStore delivers transparent persistence, eliminating any need for the object-relational mapping code that typically comprises anywhere between 25% - 70% of the coding effort in applications that use relational databases.

ObjectStore uses the native syntax of the chosen language to create, update, query, and delete persistent data objects. These objects are managed in the database with their relationships intact—in the same manner as transient Java or C++ objects. This enables retrieval of related objects through navigation of the references between the objects in the database without need of run-time mapping or joins.

Even greater performance advantages can be realized when objects, related to one another by their business or transactional needs, are clustered together to take advantage of ObjectStore's page-level data access.

MIDDLE TIER SERVICES

To enhance operations in distributed environments—particularly those utilizing application servers or multi-tier architectures—ObjectStore provides Middle Tier Libraries for both Java and C++ (JMTL and CMTL). These libraries minimize the resource demands that distributed applications place on a database by grouping a set of “virtual transactions” into a single physical transaction that accesses the database server. Such pooling reduces the demands on the database and improves application scalability by concentrating processing in the middle tier, where data is readily available to the application in the local cache. To enhance operations in distributed environments—particularly those utilizing application servers or multi-tier architectures—ObjectStore provides Middle Tier Libraries for both Java and C++ (JMTL and CMTL). These libraries minimize the resource demands that distributed applications place on a database by grouping a set of “virtual transactions” into a single physical transaction that accesses the database server. Such pooling reduces the demands on the database and improves application scalability by concentrating processing in the middle tier, where data is readily available to the application in the local cache.

THREADED ARCHITECTURE

ObjectStore makes full use of kernel threads, asynchronous I/O, and shared memory communication of the operating system. Clients use the ObjectStore's Session Management Facility to support multi-threaded applications, making it ideal for the implementation of middle-tier servers. A single ObjectStore session can have multiple threads sharing a single database transaction or one process with multiple sessions running multiple independent transactions. This provides ideal database support for the many threads of largescale application servers that must service many simultaneous requests.

C++ TOOLS

Dynamic Data Modeling Library – Offers an alternative to standard C++ Object classes, enabling changes to a data model to be performed without program recompilation or schema changes.

ObjectStore Inspector – Provides a graphical browser for managing the logical database schema and its physical attributes. Also enables the viewing and editing of the database outside of application control.

JAVA TOOLS

Java Data Objects - JDO – Developers can leverage the transparent persistent data model and enterprise scalability of ObjectStore through a standards-based Java API.

ObjectStore Java Browser – Graphically displays the cache schema to enable examination of object classes, thus facilitating data query and debug activities.

SUPPORTED PLATFORMS:*

Both 32-bit & 64-bit

- > Windows Vista, XP, Win2003 with Visual Studio .NET 2005
- > Red Hat Linux 4.0 update 4 with GCC 3.4.6
- > Solaris 9, 10 with Sun Studio 9, 10, 11
- > HP-UX 11 v2 with CC 3.63

* See the ObjectStore web site at www.objectstore.com for complete platform compatibility matrix.

Worldwide Headquarters

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA
Tel: +1 781 280-4000 Fax: +1 781 280-4095
On the Web at: www.progress.com

For international office locations and contact information, please refer to:

<http://www.progress.com/worldwide>

Progress, ObjectStore and Cache-Forward are trademarks or registered trademarks of Progress Software Corporation in the U.S. and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Any other trademarks or service marks contained herein are the property of their respective owners.

ABOUT PROGRESS SOFTWARE

Progress Software Corporation (NASDAQ: PRGS) provides application infrastructure software for the development, deployment, integration and management of business applications. Our goal is to maximize the benefits of information technology while minimizing its complexity and total cost of ownership. Progress can be reached at +1-781-280-4000.

www.progress.com

PROGRESS
SOFTWARE



0000105324