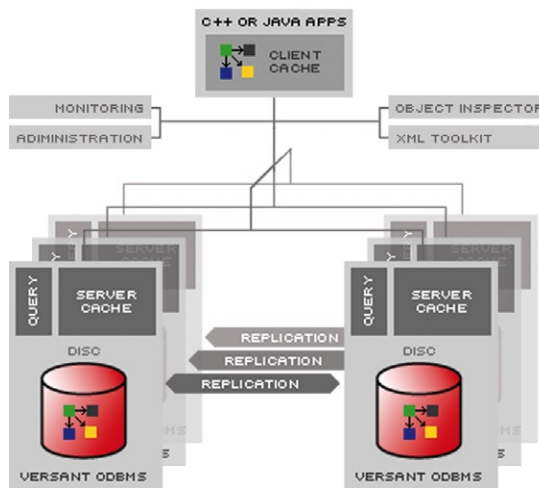


FUTURE-PROOF DATA MANAGEMENT FOR MMOG

Using the Versant Object Database for data distribution and storage brings powerful advantages to MMOG applications. MMOG's require seamless distribution of C++ or Java objects to support the vertical and horizontal scalability requirements of a multi-million user application.

The game requirements of massive concurrency, large data set support, minimum latency response, micro transactions and increasing complexity of game models, make the use of traditional relational database technology too expensive, too difficult to implement and tremendously risky. Whereas the Versant Object Database 7.0 is designed to handle parallel queries, navigational access, data distribution, and enterprise scale required by MMOG applications.



MMOG APPLICATIONS NEED A DISTRIBUTED DATABASE TECHNOLOGY

For MMOG applications, the challenging aspects of delivering a successful game are in delivering zero latency response while supporting extremely large users bases coupled with large data sets. A successful implementation needs to be able to control costs by incrementally scaling on demand in a predictable manner as the user base grows. The Versant Object Database is designed to meet these requirements by providing clusterable shard caching for the game logic tier with transparent distribution in the data tier. In effect, individual shards can actually have access to all the data in the game without data replication in the data nodes.

Clustered shard caching fulfills the need for both real-time response and incremental growth with the user base. Further, as the shard tier clusters horizontally, the data tier will naturally begin to saturate. The data tier can scale vertically by adding hardware resources or horizontally by adding more data nodes. The Versant object database is designed to provide massive scalability in a single data node through sophisticated in-memory database processing while providing reliability through replicated memory spaces and asynchronous disk I/O. Additional data nodes can be added and incorporated into clustered shard processing even allowing relationships back to common information in other data nodes.

Features

- Transparent object persistence from C++ and Java
- Support of standards J2EE & JDO
- Seamless Data Distribution across multiple platforms
- Enterprise-class high availability add-on modules
- Dynamic schema evolution
- Low (or no) administration
- Objects end-to-end architecture
- Fine-grained concurrency control
- Multi-threading, multi-session
- International character set
- High-speed data capture

Benefits

- Fast storage, retrieval, navigation of object heirarchies and graphs
- 10 x performance of RDBMS
- Cut development time

Typical Industries

MMOG, Online Gaming, Simulation, Telecommunications, Defense, Medical, Finance

New for Version 7.0

- Improved server side ad hoc query
- Increased scalability and performance
- ANSI C++ compliance

Operating Systems

Windows, Solaris, Linux, AIX, HP-UX

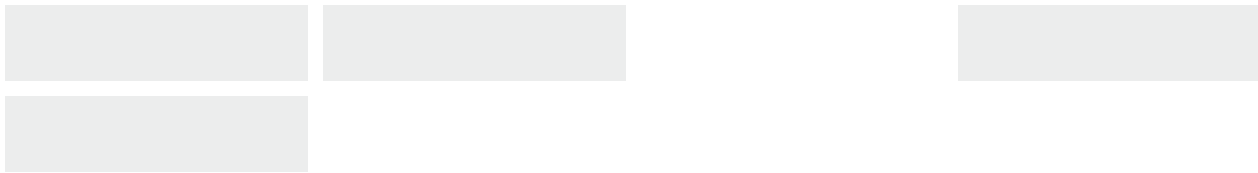
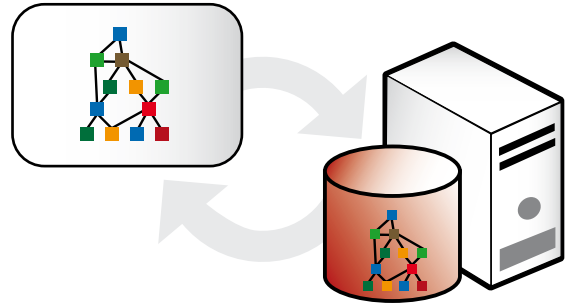
VERSANT Object Database



OBJECTS END-TO-END

Objects end-to-end means that your application objects exist in the client, on the wire, and in the database. There's no mapping or transformation of the objects between their in-memory representation and the database, unlike when using an RDBMS.

The client cache in your application transparently caches objects for speed. The database is object-aware, it can perform queries and manage indexes, allowing the application to balance processing between itself and the database. XA capability enables coordination with other transactional sources.



WHY VERSANT?

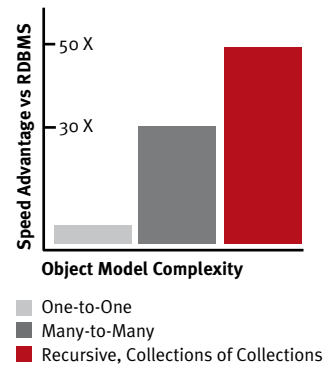
TIME-TO-MARKET: REDUCING DEVELOPMENT TIME

Considering a complex object model and usage of RDBMS, object-relational mapping code can be 40% or more of your application. It is difficult, time consuming, expensive in development, and expensive at run time, to map the objects into a relational database and – last but not least – performance can suffer. With Versant, mapping code is no longer required.

PERFORMANCE & THROUGHPUT: UP TO 50X FASTER THAN RDBMS

When applications have complex in-memory object models with predominantly navigational access, the Versant Object Database provides higher performance than mapping to relational databases. As an example, when an application needs to access information in your domain model, just like when using an RDBMS, a query can be issued to retrieve important objects. However, unlike when using an RDBMS, accessing more information related to your queried objects, does not require more queries.

Objects with moderate complexity are typically 3x faster in the Versant Object Database, objects with high levels of complexity, such as many-to-many associations are typically 30x faster when using the Versant Object Database. For collections of collections and recursive relationships, a 50x speed advantage is possible.



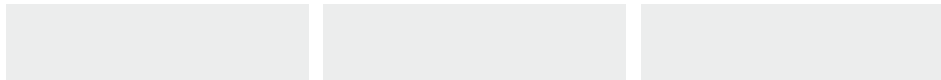
TOMORROW'S REQUIREMENTS: EASY AND QUICK EVOLUTION

Today's rate of change in process and structure and application requirements makes the ability to easily and quickly change very valuable. Object-relational mapping and other methods to adapt to rigid storage structures makes changes difficult. The Versant Object Database greatly facilitates the ability to meet current and future requirements when it comes to scalability or object model evolution.

ROI: MORE PERFORMANCE EQUALS LESS CPU'S

The Versant Object Database is the preferred solution when you have complex object models and large data sets. The main benefits are reduced code size, lower development costs, faster time-to-market, low or no administration, and reduced cost of hardware and server software licenses. The performance advantage can lower the costs dramatically for high workload applications.





SHAPE THE FUTURE. BEFORE IT SHAPES YOUR GAME.

GAME SHARD: CLUSTERABLE CACHE

Virtual world caching for minimum latency, real-time response. Game object lifecycle management is seamlessly integrated with a horizontally scalable cluster of shard caches. Each shard is capable of holding all or a portion of the complete game world. Update operations to game objects are reflected in the complete cluster through cache invalidation and automatic refresh protocols. Individual shards are multi-threaded and capable of supporting tens of thousands of users. Clustering the shard caches provides horizontal scalability to the multi-million user base of a successful game.

GAME PLAY: MINIMUM LATENCY

The Versant Object Database provides an in memory database for minimum latency transactional throughput allowing immediate return to game logic control without waiting on backend I/O operations. Versant uses piggy backed double buffering to ensure non-blocking memory writes. The minimized waits on backend structures facilitate thousands of concurrent updates per second on a given data node that is supporting a cluster of shards. A single data node is capable of supporting tens of thousands of active game clients.

GAME WORLD: PARALLEL PROCESSING

The shard cache is designed to work with distributed data nodes without requiring coding that is aware of the physical distribution. Operations, such as queries, on the data nodes are defined with logical units and carried out with threads executing in parallel within the shard and data nodes. Game objects may even have relationships spanning physical data nodes without any special code to handle the proper redistribution of game state. Changes are automatically moved to the correct data node on flush operations or transaction boundaries.

GAME STATE: HIGH AVAILABILITY

The game have the ability to define replica sites for their data nodes. Replicas are written synchronously using parallel processing to provide redundancy of data operations with minimal overhead. Failure in any data node is automatically detected and shard activity is continued to the live site with zero transaction loss. Even in flight transactions are recovered and failed sites are resynchronized to bring full synchronous replication back online.

GAME OBJECT: DISK CLUSTERING

Each data node can define physical clustering rules for logically related game objects. This means for any given data node that does not completely fit into memory, disk I/O operations will be minimized by loading logically related objects in a single disk seek. This significantly reduces wait time within the data nodes facilitating the support of larger user bases with a single data node process.

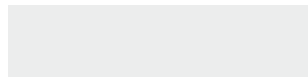
GAME SCHEMA: EVOLUTION ONLINE

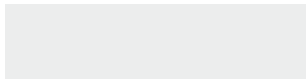
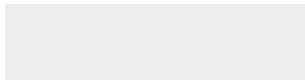
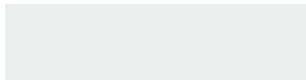
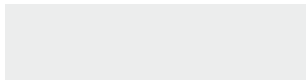
Game objects can be added and evolved online without taking the data nodes off-line or restarting shards. Game schema information is versioned and existing objects are lazily evolved on access allowing evolution of multi-terabyte databases in seconds. Existing game objects are loosely coupled with the data tier schema allowing older game objects to continue to interact with improved data nodes.

Free Evaluation copy and
Whitepaper "Advanced Data Management for MMOG"
are available for download at
www.versant.com/developer



VERSANT Object Database





ADD-ON MODULES

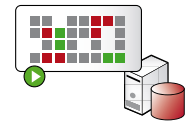
VERSANT VITNESS: ADVANCED MONITORING AND ADMINISTRATION

Using Versant Vitness for the advanced monitoring of your entire Versant Object Database environment provides you with a basic tool for all fundamental database administration operations and decisions.



VERSANT VORKOUT: RECLAIM UNUSED SPACE

The Vorkout add-on module is for applications that delete large numbers of objects. It lets you reclaim unused space in your database while it is in normal operation, increasing available free space and improving database performance.



FAULT TOLERANT SERVER: AUTOMATIC FAIL-OVER

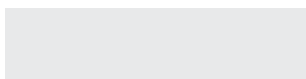
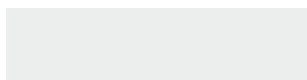
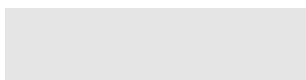
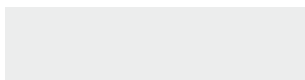
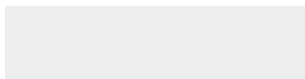
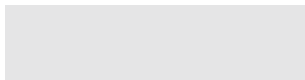
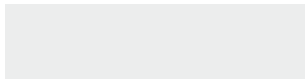
The Fault Tolerant Server (FTS) option enables automatic fail-over and recovery in the case of hardware or software failure of the Versant database. FTS uses synchronous replication between two database instances and supports transparent re-synchronization in the event of a failure.

ASYNCHRONOUS REPLICATION

The Asynchronous Replication option supports both master-slave and peer-to-peer asynchronous replication between multiple object servers. This can be used to replicate data to a distributed recovery site or to replicate data between multiple local object servers for increased performance and reliability.

HA BACKUP: ONLINE BACKUP

High Availability Backup is an option that enables Versant to use the disk mirroring features of EMC Symmetrix or other enterprise storage systems to take an online backup of very large data volumes without impacting availability.



Versant USA
255 Shoreline Dr, Suite 450
Redwood City, 94065 California, USA
Phone +1 800 VERSANT
e-Mail info@versant.com

Versant Europe
Wiesenkamp 22 b
22359 Hamburg, Germany
Phone +49 40 60990-0
e-Mail info@versant.com

