

Telecom/Network Infrastructure

An embedded database delivers fault-tolerance, real-time performance and significant manufacturer cost savings.

"We determined that integrating a proven data management solution will shorten development and testing, cutting overall development costs."

-- F5 Networks

Overview

Modern voice and data transmission depends on a chain of computing tasks executing nearly simultaneously, with the entire system always available. At the same time, equipment at seemingly every network node is adding features that must retrieve, sort and store complex data. Rules engines, network fault management, roaming service, fraud/threat detection, chassis management and revenue assurance – these are just a few of the emerging telecom and network communications infrastructure applications in which data management plays a critical role. How can developers meet this need reliably without giving up system speed and availability?

The *eXtremeDB*TM Fusion embedded database system by McObject® offers a powerful and popular solution. *eXtremeDB* Fusion's hybrid in-memory/on-disk architecture offers unparalleled flexibility. When managing data in memory, the technology eliminates disk I/O and other performance overhead; uniquely, *eXtremeDB* Fusion can even be deployed in the operating system kernel, for the ultimate in zero-latency data storage and retrieval. *eXtremeDB* Fusion's optional on-disk storage capability delivers guaranteed persistence as well as flexibility in transaction logging policies and in other software and hardware design decisions.

High Availability Database System

For always-on reliability, *eXtremeDB*'s proven High Availability (HA) sub-system enables deployment of two or more synchronized embedded databases within separate hardware instances, with automatic failover. Chosen by numerous telecom and netcom manufacturers, *eXtremeDB*'s high availability support complements the product's ACID



(Atomic, Consistent, Isolated and Durable) transactions, transaction logging and other features that ensure data durability.

Efficient Development, Cost Savings

Manufacturers also benefit, in tangible ways, from *eXtremeDB* Fusion's development ease. The product's native C/C++ API is intuitive to use, and leads to easily maintainable code (an SQL interface is also available). The database provides sophisticated debugging, and supports virtually all data types and querying methods. Deploying the proven *eXtremeDB* Fusion in place of a 'homegrown' database dramatically reduces development, unit testing and QA, slashing the time-to-market for new telecom and netcom gear.

Because *eXtremeDB* Fusion technology can be easily extended by adding new indexes and tables, and naturally separates application logic from database logic, equipment based on McObject's technology will accommodate upgrades and new features much more easily than devices using self-developed data management code.

eXtremeDB Fusion Features and Benefits

High Availability – Fault tolerant embedded database, proven in numerous commercial applications, offers the highest degree of reliability and data durability.

eXtremeDB Fusion Features and Benefits (cont.)

Kernel mode database – eXtremeDB Fusion’s in-memory data storage meets most real-time performance needs. Integrating the database in software deployed as kernel modules is also possible, for the ultimate in high-priority, zero-latency execution.

Industry-tested solution – Motorola, SOMA Networks, Hutchison, F5 Networks, NexTone, Nokia Siemens Networks and other industry notables embed McObject’s database system in their telecommunications and network communications equipment.

Hybrid data management – eXtremeDB Fusion combines in-memory and on-disk data storage in one embedded database system, to optimize applications for persistence, speed, cost and form factor.

Developer efficiency – eXtremeDB’s support for multiple data and index types (including telecom-specific Patricia Trie indexes), advanced debugging capabilities and intuitive, type-safe API boosts developer productivity, for faster time-to-market and better, safer code.

Concurrent access – in-memory databases can be created in local or shared memory. Multiple processes and threads share the on-disk database cache. A remote interface provides network access.

Application areas:

Rules engine, roaming service, network device/element database, equipment configuration, AAA (authentication, authorization, accounting), fraud and threat detection, call management, equipment monitoring, traffic analysis, routing table management, network interconnect management, phone contact database, shortcuts list, chassis management, test equipment/protocol analyzers, billing/revenue assurance, event and alarm response.

Technical Specs

- Code size from 50K to 250K, depending on the platform and features.
- Source code and object code licenses are available.
- Transaction performance measured in microseconds.
- In-memory and on-disk data storage co-exist within eXtremeDB Fusion database instances, via simple database schema declarations.
- 64-bit edition offers real-time processing of very large databases (VLDBs); in tests managing databases larger than one terabyte, eXtremeDB-64 processed 87.78 million query transactions per second (benchmark report is available).
- Implements Patricia Trie index for efficient management of IP address and routing data. Other querying methods include B-tree, R-tree, hash table and custom indexes.
- Native C/C++ and SQL interfaces; supports varied data types and sophisticated debugging features.

Architectures supported:

32-bit, 64-bit, ARM, DSP, Embedded Intel® (Pentium, Embedded Intel® Architecture etc.), Freescale (Coldfire, MCORE, HC08 etc), MIPS, Power Architecture™ (including PowerPC), x86, XScale

Operating systems supported:

VxWorks, QNX Neutrino, Linux and embedded Linux distributions (Wind River, MontaVista, LynuxWorks etc.), Windows Embedded, Mentor Graphics/Nucleus, INTEGRITY, eCos, LynxOS, RTXC Quadros, uCLinux, µC/OS-II, HP-UX, Sun Solaris, Bare bones boards (no operating system required).