



**SCHOONER MEMBRAIN™:  
THE FLASH-OPTIMIZED  
MEMCACHED CACHE and  
NOSQL KEY-VALUE STORE**

**WHITE PAPER**

*How Membrain dramatically improves the  
performance, availability, power consumption, and total cost of ownership  
of memcached software caching and NoSQL key-value stores,  
while simplifying management*

**Schooner Information Technology**

501 Macara Ave, Suite 101  
Sunnyvale, CA 94085, USA

Tel: +1 408-773-7500

Fax: + 1 408-736-4212

[info@schoonerinfotech.com](mailto:info@schoonerinfotech.com)

[www.schoonerinfotech.com](http://www.schoonerinfotech.com)

## Contents

Introduction .....	3
Memcached Overview .....	4
Membrain Architecture .....	4
Hardware Platform .....	4
The Schooner Operating Environment (SOE) .....	5
Memcached Client/Server Protocol .....	5
Data Fabric .....	5
Flash Management .....	6
Flash and Network Management .....	6
Administration .....	6
Key Capabilities .....	7
Using Flash Memory to Increase Capacity, Reduce Power, and Provide Persistence .....	7
Using Replication and Automated Failover for HA and Disaster Recovery .....	9
Using Backup and Restore to Protect Against Data Corruption or Loss .....	10
Using Containers to Manage Consolidation .....	11
Using Cloning to Rapidly Deploy and Warm Identical Membrain Nodes .....	12
Using Administrative Services to Deploy, Management, and Scale .....	12
Network Configuration .....	13
Enhanced Memcached Statistics .....	13
Putting It All Together: Membrain Use Cases .....	13
Reducing TCO through Consolidation .....	13
Industry-Leading Data and Service Availability .....	14
Summary .....	16
About Schooner Information Technology .....	16

---

## Introduction

With the explosive growth of high-volume web sites, software-as-a-service (SaaS), cloud computing, and other emerging web-based applications, datacenter workloads have increased exponentially. The business opportunities that are created from the deployment of these new applications are substantial, but the demands they place on the datacenter are daunting. Challenges include unprecedented data growth, large increases in rack space and power consumption, too much time-sensitive data to process in a rapid and consistent manner, inability to effectively scale to meet demand, underutilized resources, and severe space, power, and budget constraints.

A software cache and / or persistent key-value store is now a key part of the software infrastructure of many datacenters. Unfortunately, the traditional caching solutions (such as the widely used Memcached protocol) and key-value store products are loosely coupled and lack overall optimization. These products do not adequately address the performance, capacity, scaling, availability, and power challenges of supporting new Web applications effectively.

Huge advances in multi-core processors, flash memory, and high-performance networking offer huge improvements in performance and power at the component level. But these new component technologies are severely underutilized when datacenters integrate them at the data-access tier. Serious scaling, performance, power, networking, space, and complexity issues still remain.

To address these challenges and opportunities, new system architectures and solutions that match the workload, operating environments, and hardware are needed. Schooner Information Technology has taken a breakthrough, holistic approach to harness these powerful technology advances to create Membrain™: a breakthrough software cache and persistent NoSQL key-value store optimized to take full advantage of flash memory, multi-core processors, and high-performance interconnects.

Membrain is a high-performance, highly available, cost-effective, and scalable solution that meets today's demanding data-access needs. Relative to traditional Memcached and the other NoSQL key-value stores, Membrain delivers order-of-magnitude improvements in performance, scaling, downtime, power, and space, while cutting total cost of ownership (TCO) by more than half. The net result is more efficient and highly-available datacenter operations and the ability to create new revenue-producing business opportunities based on rapid access to terabyte-scale data.

Membrain powers production workloads in leading Web facing, SaaS, cloud, and enterprise datacenters today. Membrain offers, among other things, the following unique features:

- Industry-leading throughput, capacity, and data and service availability by tightly integrating the Memcached protocol with the innovative Schooner Operating Environment (SOE) and standard server and flash memory, significantly reducing capital expense, power consumption, rack space, and networking requirements;
- Comprehensive data and service availability through a rich set of high-availability/disaster-recovery (HA/DR) features, including Persistent Caching, Active/Active Replication, RAID, and Backup & Restore; and
- 100% memcapable compatibility with all existing Memcached client libraries, applications,

and tools, making it easy to integrate into existing environments.

Schooner customers highly value the consolidation and TCO savings afforded by Membrain. For example, using Membrain instead of traditional DRAM-only caching brought a social networking company a 10:1 reduction in the number of servers and a media company a 12:1 reduction in the number of servers needed to service the same workload. Membrain customers enjoy large savings in TCO, power, space, and application complexity.

The industry-leading data and service availability provided by Membrain is a key benefit valued by its users. Some Schooner customers use Membrain as their sole data repository, without a database as the primary data store. For these customers, the comprehensive data and service availability of Membrain is critical. For example, a Schooner telecom customer stores all the address books for users of mobile and cable services in Membrain servers. They configure the Membrain nodes with transparent replication and automatic failover, and they use the Membrain Backup/Restore facility for periodic backups for recovery from disasters and less catastrophic forms of data corruption, such as those caused by client application defects.

The rest of this white paper describes: Memcached; the architecture of Membrain; the key capabilities of Membrain; and use cases for Membrain. We conclude with a summary of key points.

---

## **Memcached Overview**

Memcached is a distributed key-value caching is widely used in high-traffic Web sites. Memcached is generic, providing caching for arbitrary chunks of data such as results from database calls or page renderings. Memcached is proliferating because of its effectiveness at off-loading database and application servers, with dramatic improvement in Web site response time and throughput.

Memcached is based on a standard client/server protocol which is maintained by the Memcached open-source community, which also maintains various client libraries, a reference server implementation, and management, administrative and compliance tools.

---

## **Membrain Architecture**

Membrain seamlessly integrates the Memcached protocol with the Schooner Operating Environment (SOE) (described later in this white paper) and the underlying hardware platform to provide superior enterprise caching and key-value store performance, capabilities, and features.

Membrain is based on a patent-pending system architecture that holistically manages state-of-the-art enterprise-class flash memory, multi-core processors, DRAM, and high-performance networking to optimize the performance, availability, and TCO of the Memcached client/server protocol. The Schooner architecture was created based on extensive workload characterization, system modeling, and optimization. The result is a highly optimized cache/key value store that effectively harnesses these powerful technologies to provide a solution that is scalable, smart, cost-effective, and green. Web facing datacenters can now leverage the advanced technology of Schooner's tightly integrated solution to quickly and easily manage data growth, decrease business complexity, and cut TCO.

## Hardware Platform

Membrain runs standard x86 servers with flash memory. The recommended hardware platform is:

- Dell, HP, or IBM x86 server or blade with at least 64GB of DRAM
- Fusion-io or other PCI-e enterprise flash memory or at least 5 SATA/SAS SSDs
- A single 10Gb or multiple 1-Gb Ethernet ports.

## The Schooner Operating Environment (SOE)

The SOE unlocks the full potential of the hardware by optimizing the utilization of all the available physical resources across a broad spectrum of workloads. It enables full utilization and linear scalability of multi-core processors, intelligently manages caching from DRAM to flash, delivers highly parallel read- and write-optimized flash memory access, and manages high performance replication of cached data between Membrain servers.

The SOE consists of the Memcached client/server protocol layer, the data fabric, the flash management subsystem, flash and network management, and administration, as shown in Figure 1.

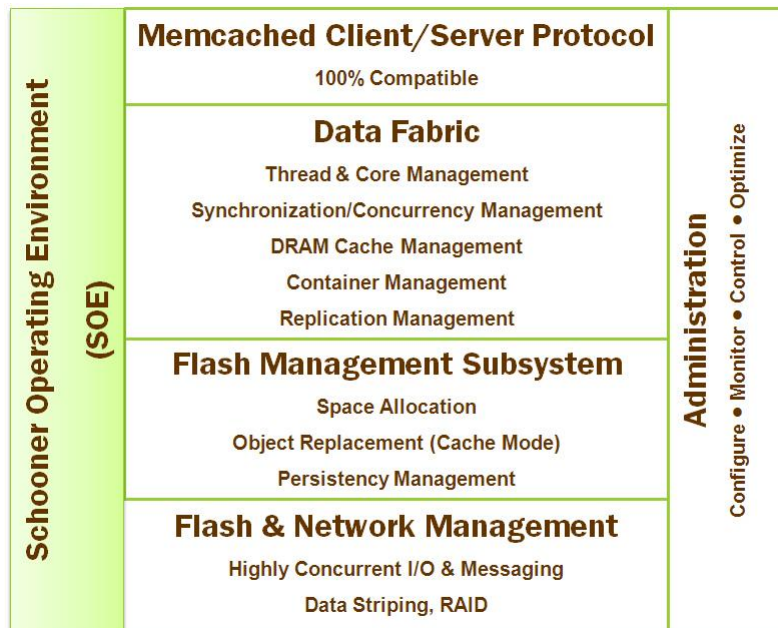


Figure 1: Schooner Operating Environment (SOE)

### Memcached Client/Server Protocol

The SOE protocol layer accepts Memcached commands and is a fully Memcapable 100%-compatible implementation of the standard Memcached client/server protocol. Membrain is therefore 100% compatible with all client applications. Using the familiar Memcached APIs, client applications can perform both distributed caching operations and persistent NoSQL key-value store operations.

### Data Fabric

The data fabric layer implements the fundamental algorithms necessary to maximize system balance, performance, power efficiency, and availability. The data fabric functions include:

- Fine-grained, highly concurrent and efficient multithreading and core management required to parallelize hundreds of thousands of simultaneous flash and network accesses per second;
- Intelligent, dynamic DRAM caching required for fast and efficient staging of both small and large objects;
- Creation and management of dynamic containers providing fine-grained control over cached data, supporting attributes of size, persistence, store vs. cache mode, and replication;
- Transparent, highly optimized data replication and failover.

## **Flash Management**

The flash management layer provides space and persistence management across the parallel flash devices. The flash management subsystem functions include:

- Mapping object keys to flash locations, employing algorithms that minimize the amount of mapping data that is kept in DRAM so that most of the DRAM can be used for caching Memcached and key-value store data;
- Maintaining persistent metadata, enabling the preservation of Memcached data across system failures; and
- Optimizing data placement and replacement across the parallel flash devices to balance accesses, maximize effective space usage, and minimize flash wear.

## **Flash and Network Management**

The flash and network management layer provides highly parallel, reliable flash-access and network-messaging facilities, optimized to exploit the full I/O capacity of the flash and networking hardware for high performance and highly available caching services. The flash- and network-management layer functions include:

- Highly concurrent, low latency network and flash access;
- RAID 5 across parallel flash devices, enabling device failure tolerance and hot swap while minimizing capacity overhead for redundancy; and
- High-performance, inter-server messaging enabling failure tolerance and live upgrade through synchronous replication.

## **Administration**

The administration services make Membrain easy to deploy, manage, and scale. The administration layer functions include:

- Simple, yet powerful centralized management facilities for easy deployment, management, and monitoring;
- Compatibility with existing tools and applications;
- Hot-key monitoring providing visibility into frequently used keys and highly active clients;
- Full and incremental backup and restore of cached data; and
- Remote presence, predictive failure analysis, and automatic restart.

---

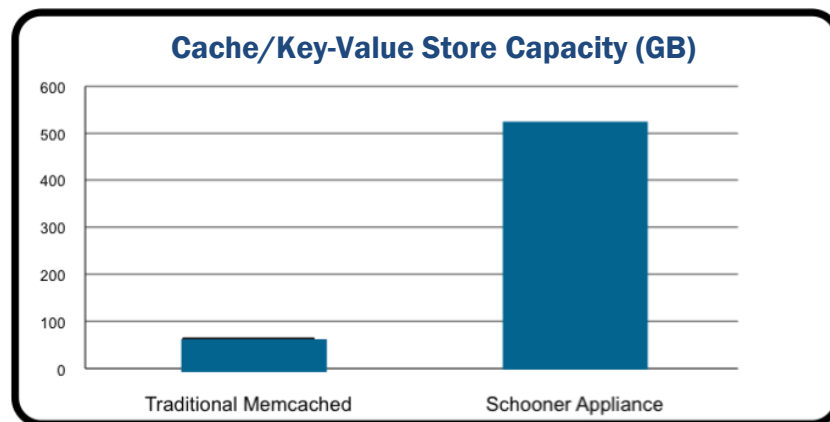
## Key Capabilities

### Using Flash Memory to Increase Capacity, Reduce Power, and Provide Persistence

Membrain integrates DRAM and flash to meet the throughput and scaling requirements of Memcached and key-value store deployments at much lower cost than a DRAM-only solution, in terms of both capital expense and power and space consumption. Schooner's flash-memory subsystem also enables the following new capabilities with Memcached:

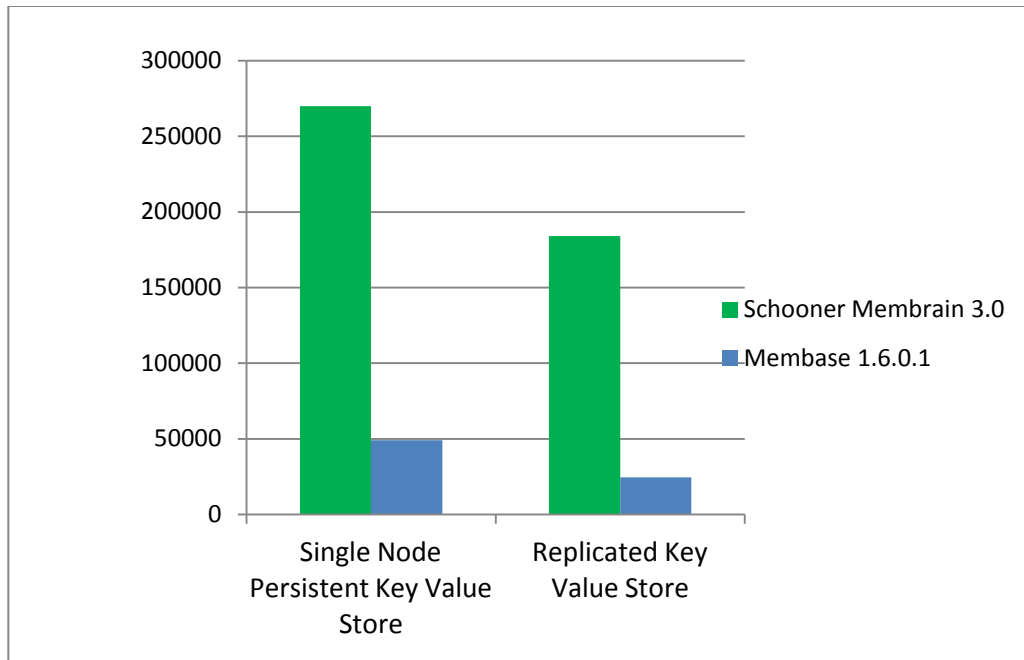
- Order-of-magnitude increase in capacity per server;
- Order-of-magnitude reduction in power consumption per Memcached or key-value store operation; and
- The ability to persist data across shutdowns/power failures.

As shown in Figure 2, Schooner's highly parallel, optimized flash-memory subsystem increases Memcached capacity by an order of magnitude over a traditional Memcached server. This is especially significant because many Memcached installations are capacity limited. For example, an application may require a total of 512 GB of memory to achieve a targeted hit rate. If a typical Memcached server has 32 GB of DRAM, this application would require 16 nodes to get the required capacity. Spreading the client workload across 16 nodes results in poor processor and network utilization (<10%). Flash memory allows the entire working set of the application to be hosted on a single node, enabling balanced use of processor and networking resources through the SOE.



**Figure 2: Schooner Cache/Store Capacity Increase through Flash Subsystem**

Figure 3 shows the throughput of Membrain relative to Membase in single node and replicated 2 node configurations. The test was performed with dual-socket Westmere servers with 1/2 TB of flash memory. The workload has an average object size of 2kB executing 90% single gets and 10% puts. Membrain achieves over 5 times the throughput in single node and replicated 2 node configurations.



**Figure 3: Throughput Comparisons of Membrain vs Membase**

Membrain delivers large consolidation relative to alternative NoSQL key-value stores, since their performance drops dramatically when data does not fit in the server DRAM. This is shown in Figure 4 which compares the single-node performance of Membrain, SchoonerSQL™ (a full high-availability build of MySQL and its InnoDB storage engine), Cassandra, and MongoDB. Each is executing the same key-value random query benchmark of 32 million and 64 million 1 KB items, on the same hardware (dual quad-core Intel processors with 64 GB of DRAM and 512GB of flash memory). The in-memory key-value random query performance of Membrain greatly exceeds other NoSQL alternatives, giving from 2 to 100 times improvement. The performance of Membrain when data cannot fit in DRAM is at least 25 times that of the alternative key-value stores.

	In DRAM	In Flash
Cassandra	10,500	1,790
MongoDB	49,000	4,000
SchoonerSQL	115,000	101,000
Membrain	310,000	160,000

**Figure 4: Throughput Comparisons of Key Value Store Alternatives**

This high consolidation significantly reduces the cost, power, cooling, and space required for Memcached or key-value store operation. As will be shown in the section “Reducing TCO through Consolidation”, the power efficiency of flash, combined with a balanced use of all system resources, results in an order-of-magnitude improvement in Memcached throughput per watt.

Since flash memory retains data when powered down, it allows Memcached and key-value store data to be preserved across system shutdowns or power failures. This can dramatically reduce the time it takes to re-populate Memcached after a restart, reducing node downtime. Clients can initiate Sync operations to Membrain to explicitly control the persistent consistency state.

## **Using Replication and Automated Failover for HA and Disaster Recovery**

In many applications, Memcached and key-value store availability is crucial to delivering the quality of service required by Web sites. If a Memcached server fails, the application must go to the primary data service, such as a database, to retrieve data. Database accesses are much slower than Memcached accesses, a problem exacerbated by the extra load when a Memcached server is down. Therefore Memcached outages can cause severe degradation in application throughput and response time. With key-value stores, the store often represents the primary source of the data. So in the event of a failure, the service will be interrupted.

Membrain solves the problem of availability in the event of planned or unplanned downtime.

Each Membrain server is assigned distinct virtual IP addresses (VIPs). Two Membrain servers can be configured as a mirrored pair, in which data stores (Memcached set/put/replace, etc.) to one server are replicated to the other. If one server in the pair goes down, the other transparently assumes its VIPs and serves its clients.

As discussed in more detail later in this white paper, Membrain supports the creation of multiple virtual storage domains called “containers” to provide fine-grained control over cached resources. Containers provide isolation and policy control, yet efficiently share processor, DRAM, and flash resources. Containers can be configured in eviction or store mode as well as persistent or non-persistent mode. Membrain replication supports all container modes.

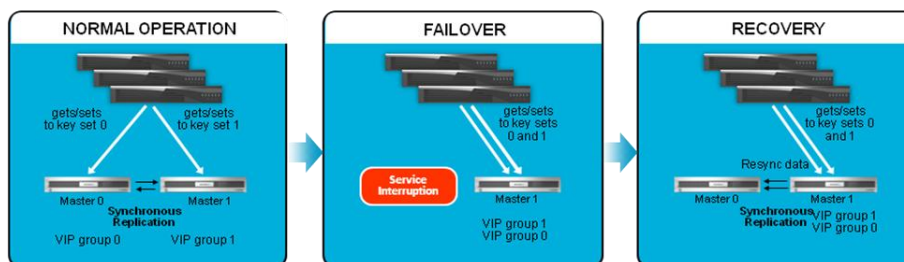
Figure 5 shows how Membrain replication works. On each write to a node, the Membrain messaging service transparently replicates the write to the other node. Replication is done synchronously so that when a client receives a positive acknowledgment for a write, the data is guaranteed to reside on both nodes in the pair. The heartbeat function of the Membrain messaging service detects failures when the messaging connection closes or times out. When this occurs, the surviving node activates the VIPs of the failed node so that it can take over servicing requests for the failed node.

When a failed replica comes back into service, it initiates actions to make itself current. First, it notifies the surviving node that it is ready to recover. The surviving node then starts copying the necessary data to rebuild the recovering cache, and also starts replication of all new write operations. When the recovering node has finished repopulating its Memcached and key-value store data, it takes ownership of its VIPs again and resumes service.

Replication is active-active: writes to a particular container can be sent to either node in a mirrored pair. Although writes to the same key could be sent to either node, this can result in inconsistent Memcached caches or key-value stores (i.e., Node 0 has different contents for the key than Node 1). To maintain consistency, applications should map distinct key sets to each node in a mirrored pair.

Membrain replication can also be used to perform a "rolling upgrade", in which the supported application is updated to a newer version without a service disruption. This is done by taking one node in a mirrored pair offline, upgrading it, and bringing it back online. When the upgraded node is back online it goes through the recovery process with the surviving node, and then resumes service.

- Active/active replication fully utilized all nodes
- Synchronous replication ensures no data is lost during outages
- Automated failover delivers continuous service availability, and eliminates the need for manual, error prone user intervention
- Transparent to client side apps and requires no additional overhead



**Figure 5: Schooner Replication**

If applicable, the SSDs in a Membrain node can be configured using RAID 5 to further increase node availability. This allows the node to tolerate the failure of a single SSD without a loss of service. When using SSDs that are hot-swappable, a defective SSD can be replaced without stopping the Memcached / NoSQL service.

## Using Backup and Restore to Protect Against Data Corruption or Loss

Many applications need to back up the contents of Memcached or key-value stores to recover from user errors or application-driven data corruption, for full recovery and restore from catastrophic data loss, for warming the caches of new servers before bringing them online, and for logging.

Membrain supports full and incremental backup and restore of persistent containers to hard disk drives. A full backup is a logical copy of all objects in a container. An incremental backup is a logical copy of objects in a container that are new or have changed since the previous backup, including a logical representation of deleted objects. A full backup is taken to start a new backup "series", which contains the full backup plus zero or more incremental backups. There is no limit to the number of incremental backups in a series. Backups can be scheduled at regular intervals.

Backups can be taken while the server is still servicing client requests. In this case, data written by a client after the backup is started may or may not be included in the backup.

Restoring a backup is the process of replaying backup streams to a server. A backup can be restored to any container. The target container must already exist and have sufficient space before the restore is started. The container can be restored to an arbitrary checkpoint within the series. See Figure 6.

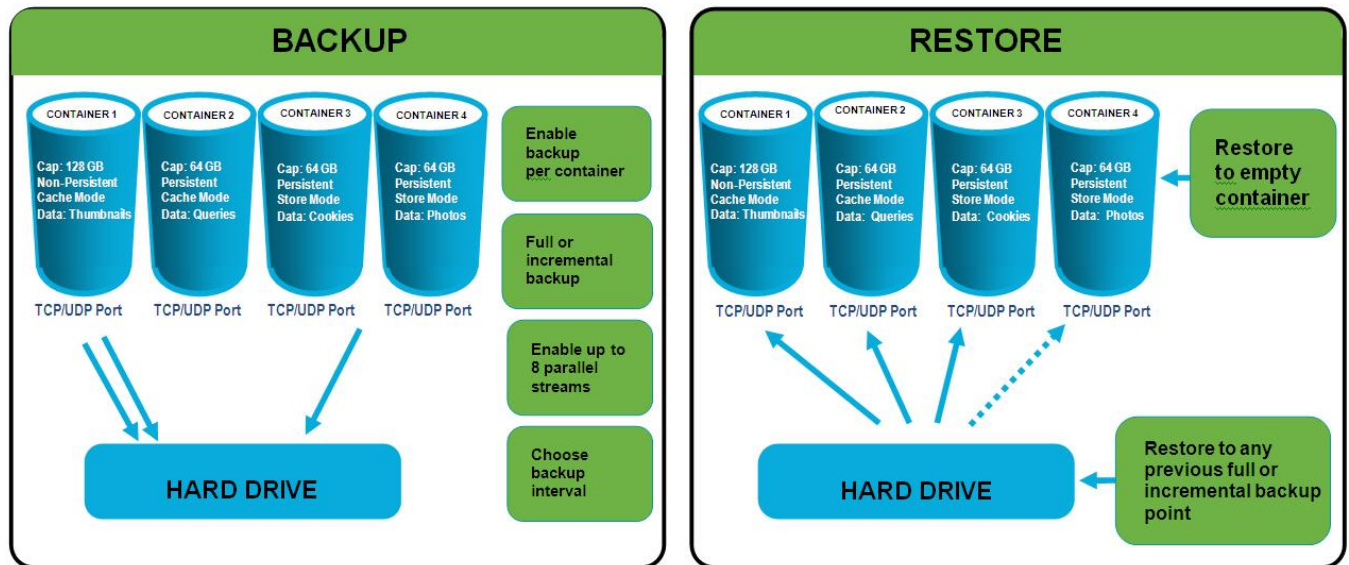


Figure 6: Schooner Backup and Restore

## Using Containers to Manage Consolidation

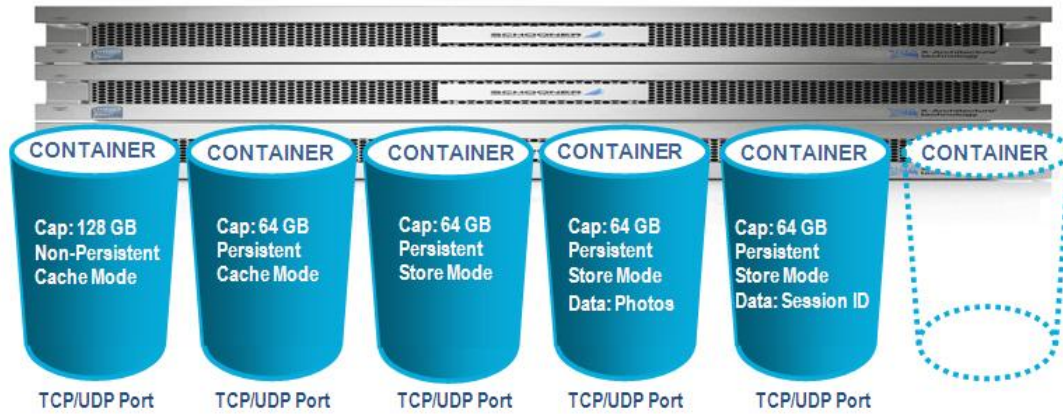
Membrain supports the creation of multiple virtual storage domains called “containers”, providing fine-grained control over cached and stored resources. The container paradigm enables:

- Isolation of separate applications or datasets based on usage patterns;
- Policy specification and control for capacity, replacement, persistence, and availability; and
- Dynamic sharing of processor, DRAM, flash, and network resources.

Containers are created and managed through the SOE administrative services, with the container attributes being dynamically enforced by the data fabric. A single Membrain node hosts multiple configurable containers. The TCP/IP addresses are shared by all containers within a Membrain node.

Each container has distinct attributes: the port through which it accepts client requests; eviction policy; storage capacity; and persistence. Each container has distinct ports for accepting Memcached requests so behaves like an independent instance of Memcached. However, since the processors, DRAM, flash, and networking are all dynamically shared across containers, the result is a much more balanced and efficient use of resources than with multiple traditional Memcached instances.

One container attribute is persistence. Using legacy Memcached, volatile DRAM is used to cache data. Membrain has introduced the ability to make cached data durable. With Persistent Caching, the cached data is retained in the flash memory and is available even after a power outage, allowing Membrain to instantly recover to its peak performance, avoiding performance-degrading, multi-hour or multi-day cache warm-up periods. Another container attribute is Store Mode. Store Mode allows client-side applications to have control over eviction of data from the cache. Data will not be evicted without the application initiating it. See Figure 7.



**Figure 7: Membrain Node with Multiple Configurable Containers**

The SOE, with its advanced processor, DRAM, flash and interconnect management algorithms, enables access to highly-available terabyte-scale flash memory at DRAM-like speeds. Persistent Caching in conjunction with Schooner Store Mode allows Membrain to be used as a persistent object store, enabling powerful new applications using the Memcached API and protocol.

### Using Cloning to Rapidly Deploy and Warm Identical Membrain Nodes

"Cloning" is the process of copying the contents of an existing Memcached server to a new one before putting the new one into service. It is an efficient way of warming a Memcached server in an installation where multiple servers hold the same data. The server that is cloned from can remain active while the cloning is in progress, with all new writes propagated to the new server. A clone operation can also be performed as a backup followed by a restore, though it may not be as efficient.

### Using Administrative Services to Deploy, Management, and Scale

Open-source tools for simplifying the administration of Memcached are very limited. That is why Membrain includes an easy-to-use graphical user interface (GUI) that allows all aspects of Memcached/NoSQL cluster configuration and maintenance to be performed with intuitive screens and menus, including:

- Network configuration;
- Adding and deleting Memcached containers;
- Adding and removing nodes to or from the cluster;
- Enabling and disabling replication;
- Restarting and recovering a failed node;
- Saving and restoring cluster configuration data;
- Backing up and restoring Memcached data;
- Querying server statistics in graphical form;
- Starting and stopping Memcached;
- Performing rolling upgrades; and
- Hot-key and hot-client statistics.

Membrain also has a comprehensive command line interface (CLI) for users who prefer to integrate Membrain into an existing administration framework. Membrain is compatible with standard cluster administration tools, including Ganglia, Nagios, Cacti, and SNMP.

## Network Configuration

Memcached installations use a wide variety of network configurations involving various forms of virtual IP addresses (VIPs), virtual LANs (VLANs), various types of interface bonding, and redundant link configuration for fault tolerance. Traditional Memcached provides no support for network configuration; the onus is on users to manually configure the networking on each server.

Membrain provides an easy-to-use interface for configuring multiple Ethernet interfaces, virtual local area networks, interface bonding (single and multiple), and virtual IP addresses.

## Enhanced Memcached Statistics

Membrain provides visibility into the state of Memcached/NoSQL servers, including statistics on:

- Hot keys;
- Hot clients;
- DRAM and flash space utilization;
- DRAM and flash access and miss rates;
- DRAM and flash eviction rates;
- Processor utilization;
- Network and flash I/O utilization; and
- Read and write response times.

Membrain provides access to Memcached server statistics via server logs and automated charting utilities in Ganglia and Nagios.

Hot-key and hot-client statistical data provides sorted lists of the most frequently used keys and the most active clients. This data is crucial for identifying client and cluster configuration problems. Membrain implements an embedded user-mode analyzer for maximum performance.

---

## Putting It All Together: Membrain Use Cases

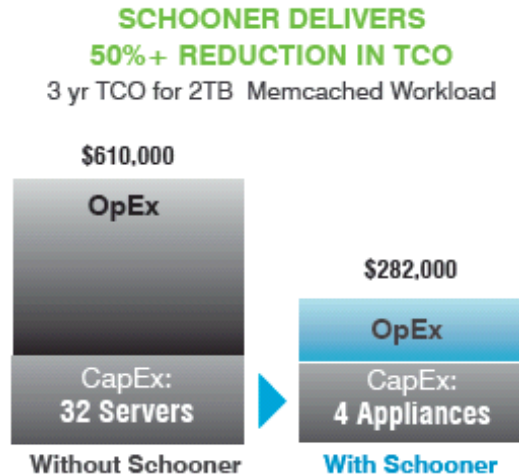
This section shows how the extended capabilities of Membrain solve customer problems in typical Memcached/NoSQL deployments through:

- Reducing TCO Through Consolidation;
- Industry-Leading Data and Service Availability.

### Reducing TCO through Consolidation

Figure 8 shows the Consolidation and Reduction in Total Cost of Ownership (TCO) which is realized in a 2-TB Memcached/NoSQL customer scenario. Consolidation provides a reduction in the number of servers from 32 to 4, coupled with a reduction in power consumption from 18 kW to

2.5 kW. TCO is reduced by 50% over three years with the consolidated Schooner solution.



**Figure 8: Consolidation and TCO Reduction with Membrain**

The TCO comparison includes initial capital expense, maintenance, and the cost of "rack, power, and pipe" over a three-year interval. The "rack, power, and pipe" estimates use rates from a typical Web hosting site. The TCO computation only includes tangible costs; it does NOT include any cost savings due to the simplified administration for an 8:1 server reduction.

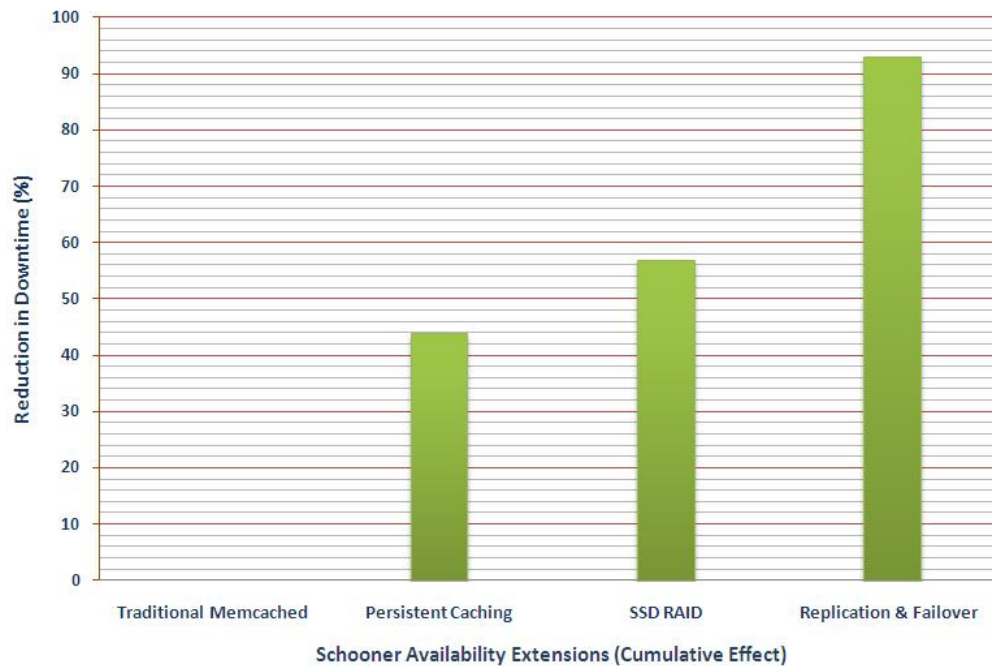
This analysis is based on a configuration of 32 traditional Memcached servers with 32 GB of memory each. It uses a standard client workload with 1-kB average object size (randomized), 128-kB average key size (randomized), 95% gets (with 1 object per get) and 5% sets. Traditional Memcached servers, limited by DRAM capacity, only use about 1 Gb of network bandwidth and only use a small fraction of the Nehalem processors. The high utilization and system balance of the Schooner architecture enable the consolidation and the power efficiency of Schooner parallel flash memory yields an eight-fold improvement in throughput per watt.

### **Industry-Leading Data and Service Availability**

Web sites increasingly depend on Memcached and key-value stores for top application performance. Failures in this tier can lead to large, unexpected loads on the underlying database, which can significantly degrade the end-user experience. Membrain provides a comprehensive set of HA/DR features to ensure a consistent user experience through both planned and unplanned downtime.

Some Schooner customers use Membrain as their sole data repository, without a database as the primary data store. Whether or not this is the case, the high availability provided by Membrain is a key benefit for many of Schooner's customers.

Membrain enables a range of availability options for Memcached and key-value store deployments. As shown in Figure 9, the Membrain availability extension can reduce the downtime more than 90% relative to a traditional Memcached cluster.



**Figure 9: Schooner Availability Spectrum**

- **Persistent Caching**

- All cached data is immediately available, even after a power outage;
- Delivers the persistence of flash memory with the performance of DRAM;
- Avoids performance-degrading, multi-hour or multi-day cache warm-up periods;
- Reduces system downtime by more than 40%.

- **SSD RAID**

- Protects from downtime caused by SSD failures and planned SSD maintenance;
- Minimizes the capacity overhead required for redundancy.
- Reduces system downtime by almost 60%, when combined with Persistent Caching.

- **Replication & Failover**

- Active/active replication fully utilizes all nodes;
- No data is lost during outages, because of synchronous replication;
- Automated failover delivers continuous service availability and eliminates the need for manual, error-prone user intervention;
- Failover is transparent to client-side applications and requires no additional overhead;
- Reduces system downtime by more than 90% when combined with Persistent Caching and SSD RAID.

- **Backup & Restore**

- Full and incremental backups can be scheduled to the onboard HDDs;

- Enables the cache data to be restored to a previous state in the event of application-driven data corruption, user error, or other data loss.

---

## Summary

In conclusion, Membrain delivers an entirely new level of enterprise caching and key-value store capability relative to traditional Memcached and other NoSQL alternatives. Membrain provides:

- **Vastly Higher Per Node Capacity and Throughput:** Eight times the capacity and throughput of traditional Memcached servers;
- **Significantly Lower TCO:** Server capex and opex reductions cut TCO by more than 50% before factoring in the value of much less downtime and administration overhead;
- **Quick Deployment:** Easy plug-and-play installation and configuration;
- **Seamless Operation:** 100% compatible with existing client applications and management tools;
- **Highest Reliability:** Delivers enterprise-class reliability with innovative persistence, replication, and recovery software;
- **Easy Management:** Simple, yet powerful centralized management and reporting with integrated CLI and GUI;
- **More Revenue:** Powers new revenue-producing applications enabled by fast access to terabyte-scale data;
- **World-Class Support:** Schooner provides 24/7/365, single-point-of-contact, Membrain service and support.

---

## About Schooner Information Technology

Schooner provides industry leading high-availability and high performance database, caching, and key-value store software for the most demanding mission-critical applications. SchoonerSQL is 100% compatible with MySQL and Membrain is 100% compatible with Memcached. SchoonerSQL and Membrain guarantee no lost or stale data, cut both planned and unplanned downtime with instant automated failover, and provide unlimited scaling with easy cluster administration. SchoonerSQL and Membrain deliver superior performance and throughput, enabling dramatic datacenter footprint reductions. With Schooner software, customers slash application downtime that costs them revenue, customers and reputation, and cut spending on servers and the power, pipe, and people to run them. Schooner is privately-held and based in Sunnyvale, CA. For more information please visit [www.schoonerinfotech.com](http://www.schoonerinfotech.com).