White Paper:

The Versatile SMP™ (vSMP) Architecture and Solutions Based on vSMP Foundation™

Revision 1.5

TABLE OF CONTENTS

TABLE OF CONTENTS **2**
ABSTRACT **3**
BACKGROUND **3**
INTRODUCTION **4**
Problem Statement (PS) 4
PS1: SMP Systems Cost 4
PS2: Cluster Complexities 4
Summary 5
Solution Framework Functional Requirements (FR) 5
FR1: Running applications that are designed for either SMPs or Clusters 5
FR2: Performance should be equal to or better than Clusters and SMPs 5
FR3: Leveraging the latest generation of chips and interconnects at any point in time 6
FR4: Management cost should match SMP deployment model 6
FR5: Acquisition cost should follow Clusters: minimizing custom hardware usage 6
AGGREGATION: NEW VIRTUALIZATION PARADIGM **6**
What is Virtualization? 6
The Versatile SMP (vSMP) Architecture and vSMP Foundation 6
Traditional SMP System Architecture 7
The Versatile SMP (vSMP) Architecture 8
Performance Characteristics of vSMP Foundation 9
Performance Benefits 9
Memory Bandwidth vs. Memory Latency 10
Closed Loop Performance Analysis 11
VSMP FOUNDATION SOLUTION SUMMARY **12**
vSMP Foundation Key Advantages 12

ABSTRACT

High Performance Computing (HPC) applications have always needed faster and larger systems, historically served by SMP systems. An alternative for large SMP systems evolved in the late 90s when x86 processors exceeded the performance of RISC processors, promoting the adoption of commodity server clusters connected with high-speed interconnects. Cluster solutions deliver more performance than traditional SMPs, but are more difficult to program, implement and manage.

ScaleMP developed the Versatile SMP (vSMP) architecture, a software-based computing-architecture that combines the advantages of shared-memory systems and the price points of commodity clusters by leveraging off-the-shelf x86 components.

This paper presents a background of the evolution of HPC solutions, a problem statement, and a solution framework for addressing these requirements. The paper provides a discussion of the overall vSMP architecture, the use of virtualization technology for aggregation and the use of advanced caching techniques to mask interconnect latencies. Finally, this paper describes how the vSMP architecture replaces hardware to create a single-system-image (SSI) platform with up to 32 processors and 1TB RAM – without modifying the OS or the application.

BACKGROUND

High performance and computational intensive applications have progressively demanded faster and larger systems to perform the work. In the late 80s, system-vendors started to offer multi-processor systems to accommodate such applications. Two types of multi-processor architectures evolved:

- **SMP - shared memory systems.** SMP stands for Symmetric Multi-Processors, but later evolved to stand for any shared-memory system, even ones that use other memory architectures, as long as all system processors can access the entire memory address space.
- **MPP - Massively Parallel Processing.** MPP systems deployed distributed memory, which required special programming techniques involved with message passing between the application fragments running on each processor.

In the late 80s, the SMP systems were the popular choice over MPP systems. The dominance of SMP was due to the ease of deployment, management and programming. However, these fully proprietary systems were expensive as they used custom processors, custom chipsets and ASICs to create high-speed backplanes as well as custom Operating Systems (OS).

In the late 90s commodity processors became faster and offered a viable alternative for server solutions. This trend accelerated in 2002 with the adoption of clusters of commodity servers interconnected with commodity high speed interconnects. These solutions were similar in nature to MPP systems and required a more difficult programming model to allow the application to span across multiple systems. MPP systems deliver more performance than traditional SMPs, but due to the need to implement and support multiple OS's and interconnect fabric technologies, are more difficult to deploy and manage.

Due to the complexity involved with cluster implementation and the cost associated with building SMP systems there is a need to provide a solution that can leverage the simplicity of SMP with the cost-structure of clusters. Leveraging x86 commodity components with software-based systems architecture can deliver the advantages of shared memory systems with the price point of cluster systems.

INTRODUCTION

Problem Statement (PS)

PS1: SMP Systems Cost

SMP systems require significant investment in system-level architecture by computer manufacturers. While SMP systems with up to eight processors can use off-the-shelf chipsets to provide most of the required system aspects, systems with more processors require significant investment in R&D, measured in the tens to hundreds of millions of dollars as well as substantial amount of development time, measured in years, to bring the solution to market.

Proprietary Systems

The result of the high R&D investment is an expensive solution that uses proprietary technology based on custom hardware and components. To date, most of the SMP systems with 8 processors or more utilize non-x86 processors, which is a large contributor to the high price of SMP systems. In addition to the high price, these systems generally "lock" the end customer into a specific computer manufacturer due to its proprietary hardware architecture and software stack.

x86 Systems

The x86 architecture was originally designed for Personal Computers, and has evolved to support servers providing low-cost server solutions with up to 4 processors, and delivering the best price-performance ratio for server systems within that class. However, the x86 architecture poses unique challenges for building larger SMP systems as this architecture lacks some core attributes required for the creation of such systems. In addition, the x86 market dynamics are such that the technology refresh cycle (processors, peripheral devices and components) is every 12 to 18 months, versus the typical time of 3 years it takes for computer manufacturers to design and build a new high-end SMP system. This mismatch creates significant risk for computer manufacturers in designing systems. It also makes it difficult to amortize and recoup the R&D investment that is required for supporting the rapid changes in this market. It is for this reason that there are very few x86 based scalable SMP systems on the market today.

PS2: Cluster Complexities

One market response to expensive SMP server systems (as described above) has been a slow migration to deployment of cluster systems using low-cost commodity x86 architecture cluster nodes with 1 to 4 processors.

Installation and ongoing management costs

However, these cluster solutions are significantly more expensive to deploy and manage compared to large server systems, requiring:

- **OS per server:** Higher OS deployment cost and complexity such as network boot or other centralized OS deployment techniques are required, resulting in a need for higher IT skill sets.
- **Solution for shared I/O:** Providing the application with access to common storage requires use of cluster file-system, SAN or NAS deployments. Achieving high-performance I/O with such solutions is still a work in progress in the marketplace today.
- **Application provisioning:** Load-balancing and distributed resource management solutions are needed to accommodate proper scheduling and resource management
- **Cluster interconnect:** A dedicated network for the intra-cluster communication is required to provide high-bandwidth and low-latency for application-level communication. This network is usually separate from the network used by the cluster to communicate with the outside world (such as users).

Programming Model

In addition to complexity, cluster deployment poses two challenges at the application level:

- **Programming model** – A specific programming model is needed to accommodate the distributed nature of the computing resource. This is usually achieved via MPI programming.
- **Lack of large memory footprint** – Each processor can access only the "cluster" node's local memory, which is usually limited in order to keep the physical size (leveraging 1U systems) and the cost of the cluster to a minimum. This poses a significant challenge to applications that use large memory in some processing phases, requiring setup of an additional system with a large amount of local memory for these application phases. This is usually referred to as 'cluster head node', and requires additional programming efforts or application provisioning techniques to accommodate the need to run different application phases on different computing resources.

Summary

While large SMP systems developed by traditional computer manufacturers carry a higher cost structure for the hardware vendor as well as end-users (PS1), their advantage compared to x86 clusters is ease of operation due to single management point as well as simpler programming model (PS2).

Problem Statement		Relative Cost	
		SMP	x86 clusters
PS1	R&D and manufacturing cost Initial acquisition cost	High	Low
PS2	Installation and ongoing cost Programming cost	Low	High

The result is a scarcity of options available for customers that want high-end systems at x86 price points.

Solution Framework Functional Requirements (FR)

The solution to the above set of challenges must address the following Functional Requirements (FR) to address the shortcomings of both traditional multi-processor systems as well as clusters.

FR1: Running applications that are designed for either SMPs or Clusters

The customer should be able to run different types of applications without the need for advanced resource management tools. Such applications might be:

- Multi-threaded
- Multi-process throughput (no messaging between processes)
- Multi-process cooperative (such as MPI applications)
- Single threaded, large memory applications

The required solutions should provide the customer the flexibility to run such different types of applications without complex reconfiguration or system setup. For example, using the same compute infrastructure for both distributed applications (needing high memory bandwidth) as well as large memory applications (needing memory footprint of hundreds of GB).

FR2: Performance should be equal to or better than Clusters and SMPs

The appropriate solution should have the ability to scale its performance across compute, memory and I/O resources in a way that will not fall short of either SMP or clusters solutions. Providing better performance compared with the traditional deployment model of specific application is an advantage.

Additionally, there are numerous advantages to having tools that help software engineers to optimize the performance of their software at the application level to leverage the solution architecture and system resources.

FR3: Leveraging the latest generation of chips and interconnects at any point in time

The solution should be designed to leverage the fast technology refresh cycle provided by the x86 ecosystem. It will allow customers to take advantage of the advance in commodity components by rapidly incorporating them to create new products in a short design cycle. It will also allow computer manufacturers to plan for and recoup their investments quickly.

FR4: Management cost should match SMP deployment model

The solution should provide for a simple initial implementation and on-going operational model – which is optimally provided today by SMP systems. Single point of management greatly reduces system management overhead and contributes to lower TCO.

The deployment of clusters requires implementation and management skill sets that are not easily found. Many customers find it difficult to handle the day-to-day IT operations required by clusters, which reduces the applicability of cluster-based solutions, in spite of the attractive initial price.

FR5: Acquisition cost should follow Clusters: minimizing custom hardware usage

The solution should provide the lowest-possible acquisition cost, best provided today by clusters.

Maximizing the use of industry standard components to take advantage of volume economics and supply will result in overall reduced cost. Reducing the initial acquisition cost is a contributor for lower TCO in addition to the day-to-day management costs.

AGGREGATION: NEW VIRTUALIZATION PARADIGM

What is Virtualization?

Computing virtualization is a technique for hiding the physical characteristics of a compute resource from the Operating System, applications or end users interacting with that compute resource.

There are two types of computing virtualization paradigms in the market today:

- **Server virtualization:** A single physical server appears to function as multiple logical (virtual) servers. It could also be defined as **Partitioning**.
- **Desktop virtualization:** The physical location of PC desktop is separated from the user that is accessing the PC. Such a remotely accessed PC can be located at home, office or data center, while the user is located elsewhere. It could also be defined as **Remoting**.

ScaleMP has created a new, third type of computing virtualization paradigm:

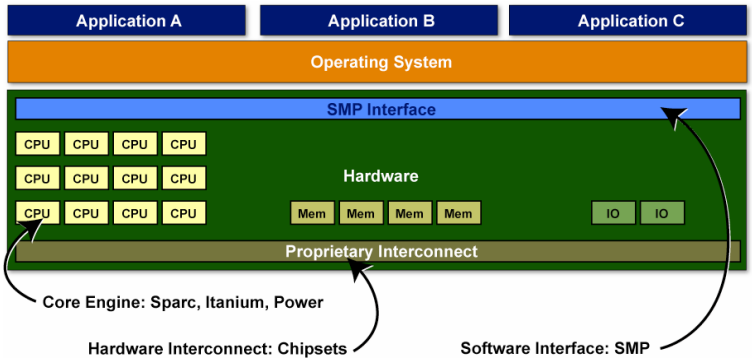
- **High-end virtualization:** Multiple physical systems appear to function as a single logical system. ScaleMP defines this virtualization paradigm as **Aggregation**, as it is basically the opposite of Partitioning.

The innovative Versatile SMP (vSMP) architecture aggregates multiple x86 systems into a single virtual x86 system, delivering an industry-standard, high-end symmetric multiprocessor (SMP) computer. ScaleMP is using software to replace custom hardware and components, to offer a new, revolutionary computing paradigm.

The Versatile SMP (vSMP) Architecture and vSMP Foundation

The patent-pending Versatile SMP (vSMP) architecture enables the creation of high-end SMP systems. The vSMP architecture fundamentally replaces the functionality of custom and proprietary

chipsets with software and commodity interconnects such as InfiniBand. It utilizes only a tiny fraction of the system's CPUs and RAM to provide chipset-level services without sacrificing system performance.



vSMP Foundation is ScaleMP's implementation of the vSMP architecture. vSMP Foundation aggregates multiple x86 system boards into one larger SMP system, allowing system vendors and value-add-resellers to create high-end x86 solutions using industry-standard components, eliminating the need for lengthy and onerous custom hardware development.

To understand how vSMP Foundation works, we will first explain the architecture of a traditional SMP system, and then dive into the details of the vSMP architecture.

Traditional SMP System Architecture

Traditional SMP systems run a single operating system (OS). The OS interacts with the system using a well-defined hardware interface, which provides the OS with predefined services to use and control the hardware. These interfaces may include hardware detection and probing, memory ordering semantics, I/O space access and interrupt delivery mechanisms. An example of such hardware interface would be the Intel's MultiProcessor Specification.

Intel describes the MultiProcessor Specification as follows:

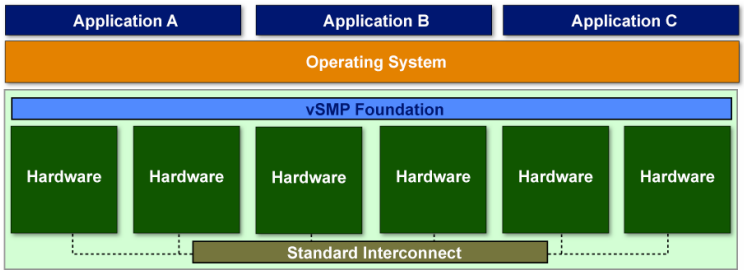
> *The MultiProcessor Specification (MP Spec) … defines an enhancement to the [x86] standard to which system manufacturers design DOS-compatible systems. … the MP defines a standard way for the operating system to communicate with the hardware. The existence of a standard interface between the hardware and the OS makes it easy for the OSVs and OEMs to quickly support a wide range of platforms with one OS version, a benefit they already enjoy in the uniprocessor desktop market for Intel Architecture CPUs. In essence, the MP Spec brings the same "shrinkwrap" benefits of the desktop market to the MP market.*
>
> *MP-capable operating systems will be able to run without special customization on multiprocessor systems that comply with this specification. End users who purchase a compliant multiprocessor system will be able to run their choice of operating systems.*

Intel's MultiProcessor Specification allows single copy of an operating system to run on a single CPU system as well as multi-CPU system with up to 255 CPUs. It details a well-defined interface that allows the OS to know exactly how to probe the hardware to determine what kind of system is running underneath it and then behaves appropriately. This interface also handles the coordination of the underlying system with the OS. For a traditional SMP system, such interface is implemented in a silicon chipset.

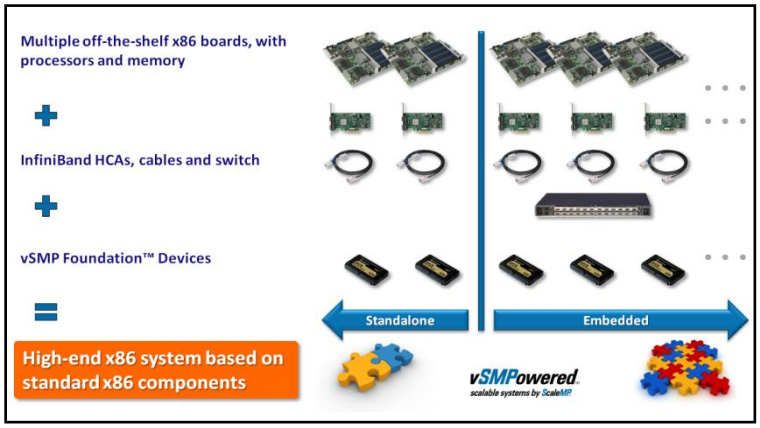
In addition to the hardware interface, an SMP system consists of CPUs, memory and I/O subsystems. These components are all connected together with a proprietary backplane or interconnect. Examples of such backplanes are Intel's FSB (Front Side Bus), AMD's HT (Hyper-Transport), SUN's CrossBar SGI's NUMALINK and IBM's XA. Such backplanes provide high-speed access between CPUs, memory and I/O – and often implemented by the chipset.

The proprietary backplane (system interconnect) is where SMP systems differ the most from each other and where the major cost of a high-end SMP system is derived. The system interconnect is expensive because the more processors that are added to a system the more complex it becomes to connect them all together in a manner that ensures

both coherency and performance. The closed architecture and high R&D costs of these systems result in systems that are highly proprietary with variants in system architecture, operating systems and applications, all driving higher costs and vendor lock-in for IT organizations.



Traditional multi-processor systems require the creation of a custom chipset to implement the system interconnect to allow processor, memory and I/O communication. The larger the system is, the more complex the required solution is. In the x86 ecosystem, chipsets that support up to 4 (Intel) or 8 (AMD) processors are available as off-the-shelf solutions. x86 chipsets that support more than 4 (Intel) or 8 (AMD) processors are complicated to design, and very few implementations exist. Moreover, as the technology refresh cycle of the x86 architecture is 12 to 18 months, chipsets and boards require significant ongoing investments to keep up with the advancement of technology. This inevitably results in slower technology adaptations in the high-end x86 market and more expensive, lower-performing systems.



The Versatile SMP (vSMP) Architecture

The vSMP architecture utilizes off-the-shelf components and does not require any custom parts. Its key value is the utilization of software to provide the chipset services that are otherwise required in creating traditional multi-processor systems. vSMP Foundation provides cache coherency, shared I/O and the system interfaces (BIOS, ACPI) , which are required by the OS. ***The vSMP architecture is implemented in a completely transparent manner***; no additional device drivers are required and no modifications to the OS or the applications are necessary.

Requirements

vSMP Foundation requires:

- Multiple high volume, industry standard x86 systems or system boards with processors and memory (processor speed and amount of memory across boards does not have to be the same),
- InfiniBand infrastructure in the form of HCA's, cables and switch (required only when aggregating more than 2 boards),
- vSMP Foundation Devices persistent storage devices (one per board) that are used to boot the system board into vSMP Foundation. The devices are plugged into each system board and are loaded with the appropriate vSMP Foundation product.

One System

Once loaded into the memory of each of the system boards, vSMP Foundation aggregates the compute, memory and I/O capabilities of each system and presents a unified virtual system to both the Operating System and the applications running above the OS. vSMP Foundation uses a software-interception engine in the form of a Virtual Machine Monitor (VMM) to provide a uniform execution environment. vSMP Foundation also creates the required BIOS and ACPI environment to provide the OS (and the software stack above the OS) a coherent image of a single system.

Coherent Memory

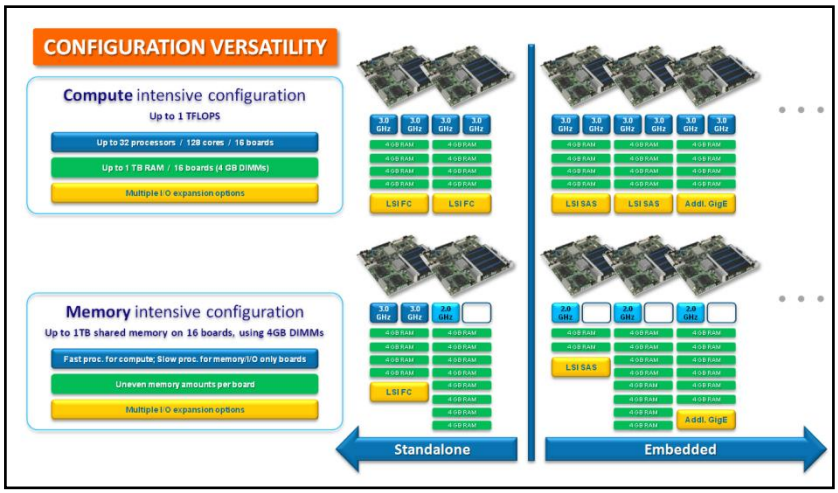
vSMP Foundation maintains cache coherency between the individual boards using multiple advanced coherency algorithms. These complex algorithms operate concurrently on a per-block basis, based on real-time memory activity access patterns. vSMP Foundation leverages board local-memory together with best-of-breed caching algorithms to minimize the effect of interconnect latencies.

Shared I/O

vSMP Foundation aggregates I/O resources across all boards into a unified PCI hierarchy and presents them as a common pool of I/O resources to the OS and the application. The OS is able to utilize all the system storage and networking controllers towards providing high-I/O system capabilities.

Versatile System

vSMP Foundation aggregates system boards with different processor speeds, varied memory amounts or dissimilar I/O devices. This is a unique capability among x86 shared memory systems.



A homogenous system with up to 32 sockets (128 cores) and 1 TB RAM, delivering more than 1.5 TFLOPS should be used for compute-intensive applications. For applications that are memory-intensive and not compute-intensive, an imbalanced configuration using both high-speed and low-speed processors can be architected. With such an imbalanced configuration, vSMP Foundation will aggregate only the high-speed processors, while not exposing the low-speed processors to the Operating System. Such a configuration allows reduced costs and power consumption, providing large-memory and top system performance. Similarly, the customer can mix and match I/O expansion options to fit application needs, making it possible to deliver the industry's most versatile and flexible high-end x86 systems. Coupled with the price/performance attributes, solutions based on vSMP Foundation provide customers the best value for their money.

Performance Characteristics of vSMP Foundation

Performance Benefits

vSMP Foundation provides the following key performance benefits:

Memory Bandwidth

The vSMP architecture enables the aggregation of memory-bandwidth across boards, as opposed to traditional SMP architecture where memory bandwidth decreases as the machine scales. This enables solutions based on vSMP Foundation to show close to linear memory bandwidth scaling. Solutions based on vSMP Foundation deliver the world's highest memory bandwidth for four-sockets and larger x86 systems.

CPU Speed

vSMP Foundation leverages the latest and greatest CPU technology available in the market at any point in time. The CPU technology refresh cycle, driven by Moore's-law, creates a faster and better CPU generation every 12-18 months. The fastest CPUs, which are also the first to market for a new generation, are the CPUs targeted at the volume server segment, which use single- and dual-processors. vSMP Foundation provides the benefits of deploying cutting-edge

CPUs at the very date those CPU models are launched, without the need to wait for a lengthy integration and productization cycle.

Reducing or Eliminating the Need for Swap

vSMP Foundation enables the creation of large memory solutions, which enables an application requiring large amounts memory to run within RAM, and reduce the need to use a hard-drive for swap or scratch space. Application runtime can be dramatically reduced by running simulations with in-core-solvers or by using memory instead of swap for large-memory footprint models.

Choice of Parallelization Paradigms

With the advent of multi-core processors, computer systems these days are using more than one core, and application developers are looking for ways to harness the added CPU power to perform more calculations in lower runtime. Some parallelization paradigms are easier to implement than others. vSMP Foundation creates true SMP machines, which enable application developers to scale and achieve shorter runtime with any of the parallelization paradigms, including threaded (OpenMP, TBB, or Explicit pthreads) and distributed (MPI, PVM, etc) codes, thereby driving performance and scalability into more applications.

Memory Bandwidth vs. Memory Latency

vSMP Foundation leverages advanced caching technologies to provide parallel access to system memory. With vSMP Foundation, data migration and replication is used to maximize system memory bandwidth. The additional system memory bandwidth is used to mask the backplane latencies. While the backplane latency of solutions using vSMP Foundation is higher than traditional SMP systems, the additional memory bandwidth offsets this higher latency. One of the keys to appreciating vSMP Foundation's ability to mask backplane latency and provide superior performance is the understanding of the fundamentals behind efficiency in memory management. At its elemental level Efficiency can be defined as

$$Efficiency = 1 - (Access \times Latency)$$

Access - The number of times a processor has to reach out to memory that is not within the processor cache (i.e. on main memory, requiring access via the backplane).
Latency - The amount of processor wait time such memory requires each time it is accessed.
Efficiency of the system can be improved by:

- Reducing the number of times the processor accesses the backplane
- Reducing the latency of each access to the backplane
- Both methods described above

Typically, the access is defined by the nature of the application, and latency is based on the technology of the backplane. Historically, the industry has improved performance by focusing significant R&D on reducing latency in each new generation of products (backplanes, memory-speed, etc). The assumption was that the access patterns were driven by the applications and hence largely out of the control of the system vendors.
ScaleMP's innovation results from its patent pending computing architecture which basically did not internally focus on improving latency (vSMPowered™ systems utilize industry standard interconnects like InfiniBand versus custom chipsets and backplanes used in traditional SMP's), but focused on reducing the number of times a processor has to access the backplane for memory operations on another physical board.
Most traditional SMP systems use Non-Uniformed Memory Architecture (NUMA). ScaleMP utilizes a combination of NUMA and Cache Only Memory Architecture (COMA) in conjunction with a massive cache (typically 5-10% of the system's RAM) and trades off backplane latency with the use of redundant RAM for caching. The backplane latency is mitigated using software-driven adaptive caching

techniques and achieving better systems economics by leveraging commodity memory costs versus proprietary backplanes and chipsets.

ScaleMP utilizes multiple memory coherency algorithms that are selected based on the memory access pattern of the application to each of the memory fragments it is using. The coherency algorithms can be grouped into several groups:

- DSRAM (Distributed Shared RAM): utilizing a 4K cache line for memory is migrated and replication. This model is greedy in nature; caching as much as possible.
- LBC (Large Block Copy): highly efficient mechanism for large data transfers. This mechanism can be used to transfer up to 128K bytes at a time. This mechanism is mostly used for remote DMA accelerator as well as instruction-based large-block pre-fetch, leveraging access pattern prediction to improve COMA efficiency.
- NASRAM (Node Attached Shared RAM): utilizes instruction-size memory access combined with fixed memory locality. This model turns specific memory areas to a non-caching NUMA model.

In essence, in spite of having higher backplane latency versus traditional SMP systems, vSMP Foundation techniques for memory access reduction more than offset the disadvantage of higher-latency, commodity industry standard interconnects and result in superior performance and scaling.

Closed Loop Performance Analysis

The major obstacle in writing scalable software for SMPs comes from the complexity involved with identifying performance bottlenecks associated with the correct use of the hardware. Different solutions allow developers to parallelize their code, but in many cases poor understanding of the underlying hardware poses a significant obstacle to archiving optimal performance. As a result, a developer must use manual optimization techniques such as use of timers in the code and educated guesses as to where the bottlenecks are located and even then cannot be sure that the top performance contention points are found.

vSMP Foundation provides a unique solution to make it possible for software engineers to clearly identify performance hotspots by monitoring application use of the system interconnect, eliminating the guesswork in performance optimization for SMP systems.

No Guesswork Performance Analysis

ScaleMP's vSMProfile™ provides unique performance analysis tools that take the guesswork out of application performance optimization. These tools pinpoint memory contention hotspots to lead developers directly to the line of code causing the performance problem.

vSMProfile captures the application use of the system interconnect by presenting the interconnect usage in a time basis graph, providing information about backplane general usage as well as usage split between the application and OS. vSMProfile provides other views of interconnect usage including interconnect usage distribution on processor or board basis, remote memory read vs. write access split and information about the memory coherency mechanism used by vSMP Foundation. Once the developer has this high-level view, vSMProfile allows him to drill down into a particular time segment of the graph to retrieve the line(s) of code that generated the interconnect usage of that segment. The data provided by vSMProfile allows the developer to see what function calls within his code are causing the most memory contention on a percentage bases, then focus-in on fixing them in priority order.

VSMP FOUNDATION SOLUTION SUMMARY

Functional Requirement	Solution Compliance	Comments
FR1: Running applications that are designed for either SMPs or Clusters	✓	vSMP Foundation provides the versatility of being able to run different types of applications at equal to or better performance compared to both clusters and traditional SMP systems (multi threaded, multi process throughput, multi process cooperative, and single threaded, large memory applications. This gives customers the ability to deploy a single system to cover all their requirements.
FR2: Performance should be equal to or better than Clusters and SMPs	✓	vSMP Foundation optimizes memory locality, providing cluster performance for distributed applications and higher memory bandwidth than SMP for applications dependent on shared memory.
FR3: Leveraging the latest generation of chips and interconnects at any point in time	✓	Solutions based on vSMP Foundation use software to replace chipset development, leveraging industry standard components, which speeds up time to market improve overall system performance and reduce system cost. System design and manufacturing cycle is reduced to less than 4 months, versus up to 3 years for a traditional SMP system or customers can chose to plug in vSMP Foundation software solution into standard systems.
FR4: Management cost should match SMP deployment model	✓	Solutions based on vSMP Foundation provide a single point of management, thus reducing the on-going operational costs compared to clusters.
FR5: Acquisition cost should follow Clusters: minimizing custom hardware	✓	Solutions based on vSMP Foundation use off-the-shelf server systems and interconnects that are traditionally being used for clusters, delivering SMP ease of use at cluster pricing.

vSMP Foundation Key Advantages

ScaleMP's vSMP Foundation makes the next generation of affordable high-end SMP systems possible for computer manufacturers and end users. High-end SMPs can be created without investing tens or hundreds of millions of dollars in proprietary R&D, and losing valuable time to market. ScaleMP enables the creation of very affordable midrange to high-end SMP computers using commodity x86 server boards and standard interconnects that deliver the lowest overall Total Cost of Ownership by:

- Run any type of HPC applications providing best of breed performance for both cluster and SMP applications.
- Use the latest generation of chips and interconnects to provide best performance at volume pricing
- Provide low management cost, by utilizing single point of management of SMP systems
- Maintain cost benefits of Clusters, minimizing the use of custom hardware and components