Dynamic adaptive parallel architecture integrates advanced technologies for petaflops-scale computing

Thomas L. Sterling

Thomas L. Sterling, "Dynamic adaptive parallel architecture integrates advanced technologies for petaflops-scale computing," Proc. SPIE 4109, Critical Technologies for the Future of Computing, (17 November 2000); doi: 10.1117/12.409225

Event: International Symposium on Optical Science and Technology, 2000, San Diego, CA, United States

Dynamic Adaptive Parallel Architecture Integrates Advanced Technologies for Petaflops-scale Computing

Thomas Sterling
Center for Advanced Computing Research
California Institute of Technology
and
High Performance Computing Group
NASA Jet Propulsion Laboratory

1. INTRODUCTION

Teraflops-scale computing systems are becoming available to an increasingly broad range of users as the performance of the constituent processing elements increases and their relative cost (e.g. per Mflops) decreases. To the original DOE ASCI Red machine has been added the ASCI Blue systems and additional 1 Teraflops commercial systems at key national centers. Clusters of low cost PCs employing COTS network technologies (e.g. Beowulf-class systems) will make peak Teraflops performance available for less than $2M in the near future for certain classes of well behaved problems. Future larger systems include the Japanese Earth Simulator with a peak performance of 40 Teraflops and three larger ASCI systems anticipated to provide peak performance of 10, 30, and 100 Teraflops culminating in 2005. These systems use existing or near term conventional technologies and architectures with some specialized integration logic and networking. While the peak performance goals can be satisfied through this strategy over the next decade, two major challenges confront the high performance computing community: 1) how to aggressively accelerate performance to the operational regime beyond a Petaflops, and 2) how to achieve high efficiency for a wide range of applications. The Hybrid Technology Multithreaded (HTMT) computer is under development by an interdisciplinary team of investigators to address both problems through an innovative combination of advanced technologies and dynamic adaptive architecture. This paper describes the strategy embodied by the HTMT architecture and discusses the key factors that may enable it to achieve two to three orders of magnitude performance with respect to today's largest systems at a cost and power consumption of only a factor of two to three times those same present day systems.

The nature and scale of a high performance computer is determined by its embodied strategy relating to how it integrates sufficient resources to enable peak capabilities, how it addresses factors that degrade efficiency, and how *ease-of-use* is achieved and to what degree. Peak capabilities include floating point performance, memory capacity, and system-interconnect bandwidth as well as other ensemble properties. The challenge is to bring enough resources together of each type to at least in principle meet the target specifications within practical constraints of cost, size, and power. For Petaflops scale systems demanding as much as a Petabyte of main memory and at least a Petabit per second bi-section bandwidth, today's technologies could easily demand 500 Megawatts of power at a cost of billions of dollars. Cooling and packaging would present enormous difficulties as floor space could exceed a million square feet.

Peak performance provides an upper bound to throughput and a lower bound to cost, size, and power but in and of itself does not determine the actual performance observed for end-user applications. While some applications have been measured at efficiencies in excess of 40% on existing MPPs, many applications experience efficiencies in the teens and some even in single digits on some of the largest conventional machines. Efficiency is sensitive to both application algorithms and system architecture. Together they contribute to four major factors responsible for the degradation of system efficiency. Overhead is the amount of work in the critical time path that is dedicated to the management of concurrent activities and parallel resources, which would not be required for a uniprocessor. Overhead actually imposes an upper bound on scalability for fixed work size computations. Latency is the distance across a system measured in cycle time of the processor making a remote service request such as a remote memory read. Access times such as these may result in appreciable delay and waiting times for the critical processor resources. Contention for shared resources may incur further delays for processors engaged in cross-system transactions. Examples include global communications channels and conflicts for access to memory banks. Lastly starvation, where a resource has no action to perform, may result either from an asymmetry in work distribution, i.e. load balancing problems, or insufficient task parallelism. Together, these factors are largely responsible for the disparity between the peak performance implied by the aggregate system resources and the sustained performance delivered during execution of

Critical Technologies for the Future of Computing, Sunny Bains, Leo J. Irakliotis, Editors,
Proceedings of SPIE Vol. 4109 (2000) © 2000 SPIE. · 0277-786X/00/$15.00

real-world application programs. How the system architecture, including hardware structure and software organization, addresses these issues strongly defines the class and nature of the architecture and should be central to strategy of the development of any new high performance architecture.

The third category of characteristics that establish the class of an HPC system is the approach to providing ease of use. This aspect of a system's nature is more vague because there are few quantitative metrics that allow measured comparisons. Nonetheless, it strongly impacts the relative utility of a system in several ways. The generality of a system determines the range of applications and algorithms for which it is suitable. Interestingly, suitability might be measured by efficiency: those applications/algorithms for which a given system performs at or above a specified level of efficiency (ratio of sustained to peak performance). But often, programs are run on systems that deliver surprisingly low efficiency. Therefore, such thresholds are subjective and more often or not are determined by which of a set of machines in one's available arsenal is likely to do the best job; not necessarily a good job. Programmability is another aspect of ease of use which is difficult to measure. The amount of time it takes to develop and debug a parallel program on a given machine might be used as a metric. But the measure varies widely among programmers and types of programs such that it is difficult to quantify programmability for any given machine. For large parallel and distributed systems, programmability is made more difficult by the complex parallelism representation and the direct resource management responsibilities. On the largest systems (e.g. ASCI Blue Mountain), several levels of distinct forms of parallelism may be employed to employ the fullest extent of the machine. These "constellation" (see Top500 list) systems cluster large distributed shared memory nodes, each comprising more than a hundred microprocessors that internally exploit fine grain parallelism. Thus, such systems require the direct use of at least three separate kinds of parallelism: message passing parallelism at the global inter-node level, coarse-grain thread level parallelism within the DSM nodes, and instruction level parallelism including VLIW internal to the processors themselves. Programmability is strongly effected by the need to explicitly craft a complex application code using all three levels. Even when the microprocessor compiler automatically detects the lowest level parallelism, the programmer often has to be very sensitive to the effect of his/her programming style on making available such parallelism and on the reuse of the multiple layers of caches. Programmability is also strongly determined by the degree to which a programmer must explicitly manage the partitioning and allocation of program data and tasks. The need for programmer control is to provide good locality of access to minimize both inter-node communication and memory access latency. Systems that automate much of this process or through architecture simply do not require some or all of these responsibilities to be performed by the programmer may be considered more programmable. Debugging and reliability contribute to ease of use as well as tools to manage parallel file systems.

This paper briefly introduces an alternative strategy to addressing all three major factors bounding the constraint space for the design of future ultra-scale computers. The HTMT, or Hybrid Technology Multithreaded approach, may enable systems capable of Petaflops scale computing delivering efficiencies greater than 50% across a wide range of application and algorithm types which are both more general and easier to program than conventional MPPs of a fraction of the performance. HTMT employs a mix of advanced technologies to provide the peak capabilities required of a Petaflops scale systems with a parts complexity of within an order of magnitude that of today's Teraflops scale systems. HTMT employs an innovative architecture supporting a dynamic adaptive resource management scheme for low overhead and automatic latency handling to deliver high efficiency. HTMT employs a hierarchical multithreaded execution model in a shared memory system which in combination with the automatic resource management provides a general and highly programmable system class for exceptional *ease-of-use*. HTMT is the product of a long-term research program to explore alternative methods to achieving ultra-scale computing systems in the next few years providing superior operational properties to today's HEC systems. The remainder of this paper describes the approach taken by HTMT to provide dynamic adaptive resource management for high efficiency and ease-of-use.

2. EMERGING TECHNOLOGIES

The physical nature of an HTMT system is determined, in part, by the technologies it employs and by the structure and organization of its components and subsystems. HTMT breaks new ground in harnessing the potential and opportunities of emerging technologies that exceed in one or more ways the capabilities of conventional semiconductor based systems. Practical systems of Petaflops scale will require order of magnitude improvements in speed, storage, and bandwidth. This section describes several such technologies.

2.1 Superconductor RSFQ Logic

The fastest logic technology is based on superconductor electronics. Logic switching speeds in excess of 700 GHz have been achieved in the last couple of years by a team at SUNY Stonybrook in laboratory demonstrations. Even today, commercial superconductor devices such as analog to digital converters can operate at 20 GHz or more. This is far in advance of early work performed at IBM in the 1970s and in Japan in the 1980s. Using Josephson Junctions (JJ), logic families were devised similar to transistor logic of the time in which logical 0 and 1s were represented by voltage levels. The resulting logic was relatively slow sustaining between 1 and 3 GHz clock rates. Because of this, the projects were not continued. In the mean time, another approach to implementing logical functions with JJs was developed initially at TRW and Moscow State University. What became known as *Rapid Single Flux Quantum* logic or RSFQ used a basic device called a SQUID comprising two JJs and an inductor in a loop. SQUIDs had been used in very sensitive instruments because of their high signal to noise ratio. Such devices can hold a circulating current indefinitely due to the zero resistance phenomenon of superconductivity. But the resulting flux can only exhibit discrete levels due to quantum mechanical effects. This makes SQUIDS ideal as basic building blocks for logic using distinct flux levels as logical 0 and 1s. From this lowest level of device, families of logic have been devised with differing properties and have been demonstrated in the laboratory. Using Niobium technology on a silicon substrate and Aluminum Dioxide dielectric, complex logic circuits can be fabricated by means of methods little different than manufacturing the metal layers on conventional integrated circuits. In fact, in principle, the manufacturing of RSFQ chips is easier than today's CMOS devices. Three commercial Niobium RSFQ fabrication lines are in operation along with several experimental lines in laboratories. While experimental RSFQ chips with feature size below 0.4 microns have been implemented in the laboratory, full commercial chips of 1 cm on a side are fabricated at between 1.5 and 3.0 microns. This is an order of magnitude behind the state of the art in CMOS fabrication and reflects the limited commercial applicability and narrow niches to which this technology has so far been directed. The largest RSFQ chips have integrated a few thousand gates at most.

There is another important property of superconductor RSFQ logic: its low power consumption. A typical example is a logic gate operating at 100 GHz fabricated in 0.7 micron Niobium technology. The power consumption for that device is about 0.1 microwatts. This is substantially less than for the corresponding functionality in CMOS. Although the cooling efficiencies for refrigeration capable of creating the required temperature regimes of 4 Kelvins is less than 1%, the resulting "wall-plug" power is still significantly less than conventional technologies. This is particularly true for advanced semiconductor devices with clock rates above 10 GHz such as SiGe. As an example, design analysis of a Petaflops scale parallel processor (excluding main memory, etc.) determined that power consumption within the cyrostatic environment would be on the order of 150 Watts.

The major challenge in designing processor architectures based on superconductor RSFQ logic is the implicit increase in latency measured in clock cycles resulting from the two orders of magnitude increase in clock rate with respect to conventional CMOS microprocessor technology. Due to time of flight at 100 GHz clock rate, the signal distance in one 10 Picosecond clock cycle is a fraction of a millimeter. A full-scale chip of 1 centimeter or more on a side is therefore ten to a hundred cycles in breadth. At these speeds, even an execution pipeline interlock required to manage potential hazards is impossible as the signal distance is greater than that of a single cycle time. One small convenience related to low-level design using RSFQ logic is that each gate has as an intrinsic property an internal latch. This allows easy implementation of super-pipelined structures without incurring the additional complexity costs of separate buffers between each stage. But the more fundamental problem of devising latency insensitive structures that contend with issues such as hazards, data access times, synchronization, and scheduling may involve advanced architectures. Systolic arrays are one method that can be employed for a narrow range of algorithms such as digital signal and image processing. Dataflow architectures may hold some promise, although perhaps different from those experimental machines (e.g. Monsoon) of the previous two decades. As will be discussed, the HTMT strategy employs a third approach: fine-grain multithreaded architecture.

The future of superconductor RSFQ logic technology is uncertain. The major technical challenges in bringing this technology to mainstream high performance computing have either been resolved or are well understood with effective approaches being pursued. For practical purposes perhaps the most serious is the disparity in feature size. Ironically, while clock rates of a factor of a hundred may favor RSFQ logic, CMOS is capable of packing approximately that much more logic on to a comparably sized integrated circuit. All of the needed fabrication equipment exists, most taken directly from the semiconductor industry. However, the necessary procedures have not been devised, calibrated, and matured to produce high yield runs of 0.25 micron Niobium. The necessary capital will have to be invested to make achieve this. Whether or not a combination of government and industry funding can be applied to this objective is not yet clear. The future of RSFQ will depend on it.

2.2 Fiber Optic Data Communications

The Achilles heal of high performance computers is the internal system area network. Where numeric intensive computations could require as much as a few bytes per floating point operation resulting in a greater than a Bps per flops bandwidth, some system's global bisection bandwidth can be as much as two orders of magnitude lower. As systems scale from the Teraflops regime to Petaflops communication requirements may grow by three orders of magnitude for some applications. Significant improvements have occurred in recent years for data transmission through wired interconnects. 500 Mbits per second is used in a number of short distance contexts with 4 Gbps accomplished in the laboratory using differential pairs and advanced active communication methods. Data rates are likely to continue to be enhanced over the next few years approaching 10 Gbps per channel. Even with these accomplishments, the need for far greater aggregate bandwidth is imposed by ultra-scale computing systems as well as by very fast processors. Interconnect complexity grows super-linear with respect to number of nodes. Use of very high speed processors such as those considered implemented in superconductor RSFQ technology reduce the number of required nodes for a given peak performance and therefore the network complexity as well. But the requirement for higher bandwidth strains the capability of wired communications.

An exciting alternative, not yet employed in high-end systems, is optical communications. Light carriers provide the opportunity for dramatic increase in data handling capacity. Two classes of optical interconnects are 1) free space, and 2) guided. As implied by the name, free space optical communications sends data modulated light signals through space (including air) physically directed to a specific point where a receiver is placed. Addressing of different input ports involves the physical deflection of the light emitter beam, either directly or through modulated reflectors. Free space optical interconnects are employed primarily as cross-bar switches; supporting all to all connections. A major advantage is the lack of physical channels such as fibers and the need for many physical fiber-end connections. One disadvantage is the need for global arbitration of the network to establish the next interconnect pattern. This usually involves wired-signaling and can incur appreciable delay due to the latencies involved. Other challenges exist as well related to registration of signal targeting in highly packed receiver arrays. To date, free space optical networks have not been employed in computer systems. However, in the long term, their advantages in parts complexity may make them the medium of choice for the very largest systems of the future.

Guided optical network technology employs fiber optics to transport data modulated light from transmitter source to receiver destination. Fiber optics has been employed for years as long-haul voice and data channels by the telecommunications industry. Recent advances in fiber optic technology have made it useful to serve in the capacity of system area networks for very high performance computers in the near future. One major advance is the combination of Time Division Multiplexing with Space Division Multiplexing to provide exceptional bandwidth. In the last few years, the number of separate wavelengths that could be simultaneously inserted practically on a single fiber channel has grown from approximately 8 to 256 as demonstrated in the laboratory. While early fiber optics operated between 2 to 4 Gbps per wavelength for a given fiber channel, data rates of 10 Gbps are possible, assuming the drive electronics at both the transmitter and receiver end can keep up. These advances in per channel bandwidth and wavelength multiplexing are delivering experimental systems capable of transporting data at a rate on the order of a Terabit per second compared to wire capacity of a few Gbps; at least a two order of magnitude gain. Another advance is in the area of optical signal switching. Early mechanical deflection mechanisms operated at switching speeds on the order of a millisecond, far too slow for the fast data packet throughput required for high performance computers. New devices based on both reflection and incidence of refraction through rapid nonlinear changes of material properties in response to electrical voltage signals have produced devices capable of operating at switching speeds of a 100 MHz with the promise of nanosecond switching times in the near future. Also in the critical path of direct application of optical data communication within computers is electro-optic integration. For parts complexity and cost of large switching optical networks to be practical, what was once accomplished by discrete devices on a large optical laboratory test bench must be integrated into single modules such as MCMs or even single chips. Recent advances in electro-optic integration make the near term integration of hybrid nodes comprising both high speed electronics such as GaAs or SiGe technology with optical interconnections, switches, sensors, and laser diodes. It should be possible to place a few such nodes on a single die within the next couple of years for mass production.

2.3 Holographic Storage

Data set size varies dramatically among classes of application and computer system performance. The rule-of-thumb that dictates 1 byte of storage for every flops of performance is too simple to model the richness of executing problems. Indeed, there are such applications and a system capable of a Petaflops of performance does require a main memory with a capacity of a Petabyte of storage. Other application types may require a small fraction of this however. Memory systems of the future may comprise multiple memory levels for the main memory. Semiconductor DRAM provides excellent price-capacity with densities anticipated to increase by a factor of four every three years or so. Nonetheless, for ultra-scale computing, DRAM memory alone could dominate overall system cost and possibly power consumption if employed to meet the full capacity needs of the worst-case applications. The possibility of employing alternative technologies to meet the capacity requirements for the largest systems while exhibiting substantially lower cost, parts complexity, and power consumption than semiconductor DRAM, therefore, can be an important enabler of such systems.

An emerging technology that holds much promise for advanced memory systems is optical holographic storage. Optical storage has become the primary physical data archive and transfer medium in the form of CDs and DVDs for music, video, and data. Initially read-only, optical data can now be written in at least write-once mode. But the speeds for these media are on the order of that of secondary storage and not appropriate for primary storage application. Holographic storage techniques including *photo-refractive* and *spectral-hole-burning* provide alternative ways of storing data in optically sensitive materials. Today, with existing devices in the laboratory, between 1 and 10 Gigabits can be stored in a single photo-refractive medium of a cubic centimeter. Future capacity may extend to 100 Gbits or more for photo-refractive techniques and beyond a Tbits for the more speculative spectral-hole-burning method. While of significant long-term interest, spectral-hole-burning is highly experimental and it is probably premature to contemplate its near-term impact. Photo-refractive methods are well understood; prototype storage systems have been successfully implemented, and the technical challenges to bringing them to practical commercial use are being addressed.

Optical storage devices such as these store data in large arrays of bits, all accessible at once. A typical bit array or page is 1 Mbits with larger pages both possible and desirable. Initially, such devices were accessed by means of mechanical reflectors. Small perturbations to the mirror angle would result in a different page being acquired. But such techniques were slow with access times on the order of a millisecond or more. This is fine for applications in the domain of secondary storage but not for main memory. Acousto-electric deflection causes small variation in the surface of some crystals through the application of high voltage signals. Access times on the order of 100 microseconds may be achievable through this means although the practical problems of generating and applying the high voltage signals can detract from the benefits of this technology. It is also slower than would be desired. Pixel arrays provide a solid state approach to this problem. A two-dimensional array of laser diodes integrated on a single semiconductor chip, each directed at the optical target at a slightly different angle will extract a unique memory page. Tunable lasers provide a different access mechanism with different wavelengths corresponding to different pages. Tunable lasers provide the advantage of requiring only a single laser. These last two techniques may yield access times on the order of 10 microseconds. Although the response time to an access request is on the order of two orders of magnitude greater than that of DRAM, the bandwidth is greater as well as the capacity by about a factor of ten in each case. Power consumption is also appreciably lower. A hybrid main memory system can be anticipated combining both DRAM for fast access with optical storage for high capacity and throughput. Together they may achieve the virtual large memory system that the use of demand paged data to disk never effectively accomplished.

2.4 Processor-in-Memory Semiconductor

DRAM technology and CMOS logic technology, although both semiconductor based involve significantly different processes. However, recent advances have allowed both classes of devices to be implemented on the same die through innovative fabrication techniques. Direct access to the output signals of the DRAM stack sense amps by CMOS logic opens new opportunities in computer system design and operation. Perhaps the most significant advance is direct access to all of the bits acquired through an initial row access. As many as 2K bits may be stored in the row buffer during a DRAM read cycle. Generally, the majority of these are lost during external memory access requests. But on-chip, they can be used through this new technology. This is equivalent to 64 words of 32 bits each read at a data rate equivalent to 32 Gbps bandwidth from a single chip. By partitioning the memory on a chip into sub-arrays, this throughput can be increased to as much as 0.5 Tbps for next generation DRAM. Other advantages include lower access latencies and lower power consumption. For those accesses processed on-chip, and where ALU logic is placed next to the row buffers, latency of access is very low; perhaps four times lower than conventional systems. Wide ALUs accessing all bits of the row buffer may perform many basic operations simultaneously where appropriate. Operation rates on the order of several Gops per memory chips are realizable with near term technology. By performing these operations on chip, the same data is not forced to go through the I/O pins of the memory chip to external processors saving both time and power, as well as avoiding a possible series of bottlenecks.

3. A HYBRID TECHNOLOGY ARCHITECTURE

The Hybrid Technology Multithreaded (HTMT) architecture is a shared memory multiprocessor structure with tight coupling between processors and memory. HTMT incorporates and integrates components implemented with the advanced technologies described in the previous section to achieve substantial advantages in performance, power, complexity, space, and cost with respect to conventional technologies and approaches. HTMT employs a memory hierarchy with small high-speed buffer memories near the very high-speed processors and high capacity memories comprising the main memory subsystem. HTMT differs markedly from conventional systems in that all memory subsystems are in the name space of the processors, and many of the memory layers incorporate their own local management processors through PIM technology. Together, these distinctions drastically alter the way processors and memory interact, resulting in very high efficiency. This section briefly describes the overall structure of the HTMT system which is illustrated in Figure 1. The following section will then discuss the means and dynamics of its subsystem interactions.

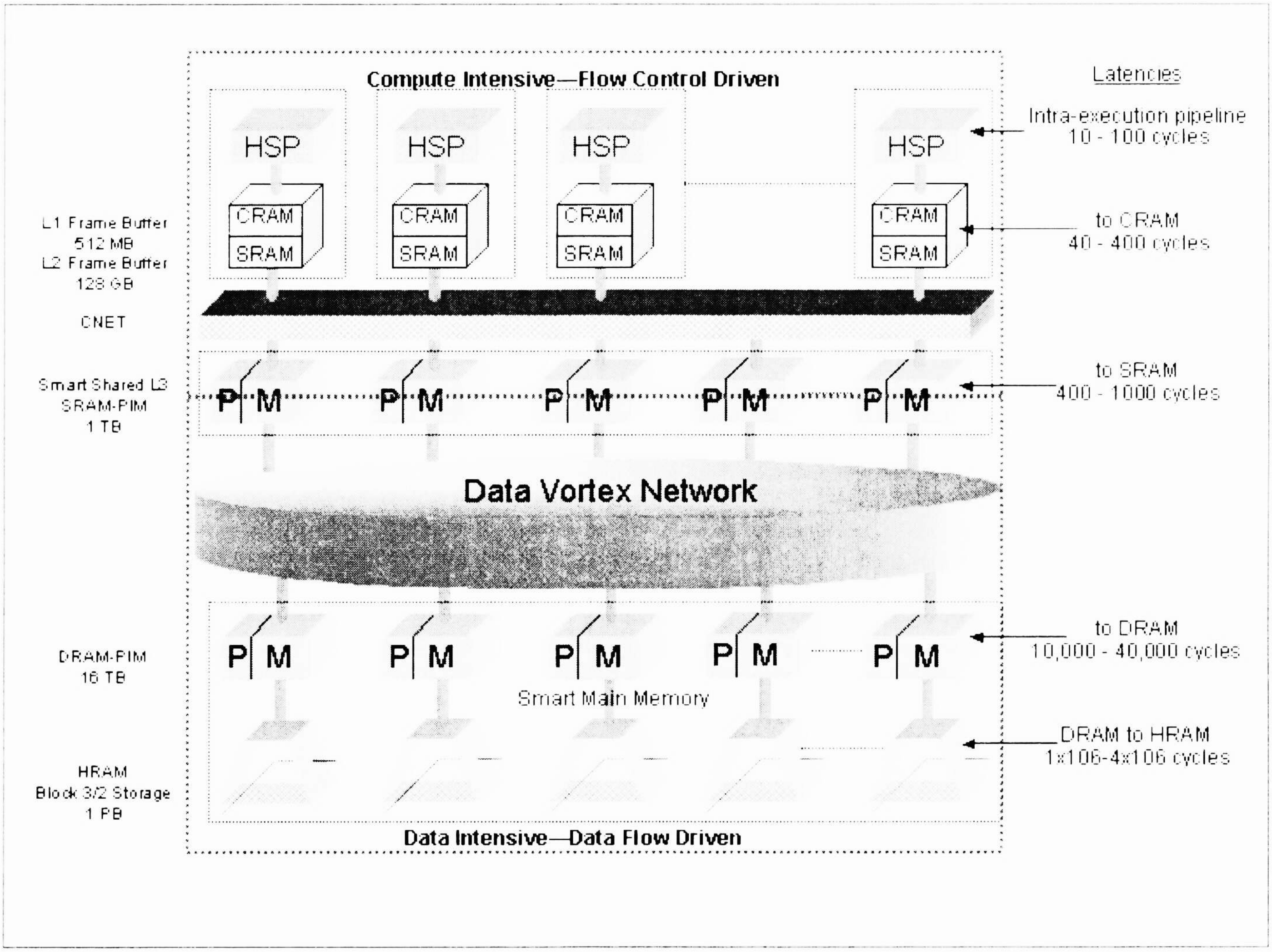

Fig. 1 HTMT-class Architecture Block Diagram

3.1 Overall HTMT Structure

The HTMT architecture comprises a processor subsystem and a memory subsystem. Yet, the processor subsystem incorporates storage elements and the memory subsystem incorporates programmable logic functions, blurring the conventional distinction between processor and memory. The HTMT architecture design point is to achieve a Petaflops peak performance with an internal storage capacity of 1 Petabyte.

3.2 High Speed Processor Subsystem

The main processor array employs ultra-high speed multithreaded processors. These have direct access to two levels of very high-speed buffer memory for holding stack frames and instruction sets. A high-speed network connects the processors and their buffer memories to the PIM-based SRAM which is the interface to the memory subsystem.

The HTMT processor is a two stage multithreaded computer capable of issuing up to four floating-point operations per cycle. As many as 4096 processors are integrated in to a single system, delivering a peak performance of just over 1 Petaflops. The processor is implemented in 0.8 micron Niobium superconductor technology cooled to 4 Kelvins with an internal clock rate of 100 GHz. The logic is derived from RSFQ as developed at SUNY Stonybrook and will be implemented on a TRW fabrication line with a new process not yet in place. A small number of these chips are packaged in an MCM with inter-chip communications within an MCM or 30 GHz per interconnect. Data clock rates between MCMs is 10 Gbps.

One system design has a processor associated with a new high speed RSFQ pipelined memory system to be developed by Hypres. At the specified clock rates, access time is dominated by time-of-flight latency. A pipeline structure is being adopted whereby more than one access may be processed by the memory at a time to increase the throughput substantially above the limitations of a single-issue memory. A possible second level high-speed buffer memory using a hybrid technology approach may be employed in the structure as well. This would employ semiconductor memories which while cooled, would not be superconducting. Even so, its speed would be significantly increased due to the lower temperature.

A superconductor network is employed to connect the processor and buffer memory ensembles together and to the external connections to the rest of the system. While the final configuration has yet to be worked out, it is likely that an asymmetric hybrid technology external interconnect will be used. Data signals entering the cryo-container will use an optical communications medium and output signals will use wired channels. Typically for many algorithms, much more data is read than is written by the processor which would permit this asymmetry. The motivation for using wires instead of optical for the output is to avoid the use of laser technology in the supercooled regime. The entire Petaflops-scale computer array will fit in to a cylinder less than 2 meters in diameter and 2 meters high. Internal power within the cryo-container will be less than 250 Watts (that is not a typo!) total.

The processor subsystem interfaces to the memory subsystem by the intervening PIM-SRAM to be described below. Most of the accesses by the processors are to this layer or to the dedicated high-speed buffer memories associated with each processor. The number of threads per processor is selected to overcome and hide the latency to the PIM-SRAM level so that accesses to this can be performed without efficiency penalty.

3.3 Memory Subsystem

The memory subsystem is dramatically more sophisticated than those found in conventional multiprocessors. This is due to the extensive use of small wide-word processors incorporated within the memory chips themselves. The memory system has four stages: main memory DRAM, PIM-SRAM, Data Vortex optical network, and holographic “3/2” memory.

The main memory consists of DRAM augmented with on-chip processors. If fabricated with today’s technology, a PIM chip used in HTMT (called the “MIND” chip for Memory-Intelligence-and-Network-Device) will contain 8 Mbytes of DRAM partitioned in to four groups, each group controlled by its own special purpose processor. The four MIND internal nodes are connected by a pair of buses which also provides the data path to shared floating-point ALUs and external I/O message ports. All DRAMs are in a global name space and may be accessed by any processor although such direct interactions are minimized. A total of 16 Terabytes of DRAM is used for the main memory layer.

A second layer of storage employing the holographic photorefractive technology is to be used as backing store for each of the DRAM groups. As much as a 100 Gbytes per module is sought to provide a total of 1 Petabyte of storage. This layer is referred to as the "3/2" storage because it is midway (logarithmically) between primary and secondary storage when measure in latency time. Access times are assumed between 10 and 20 microseconds with pages of 2 Mbits acquired per access. Bandwidth is comparable or superior to the DRAM layer.

The PIM-SRAM layer is the memory subsystem's high-speed buffer array and used to interface with the processor subsystem. It too incorporates on-chip processors but with SRAM rather than DRAM cells. Two processors are incorporated per chip. Although this layer plays the same role as an L3 cache in a conventional system, it is in the name space of the machine and can be accessed by any processor or other PIM chip.

The PIM-DRAM and PIM-SRAM layers of the memory system are interconnected by a high-speed optical interconnect system called the "Data Vortex". This network uses both time division and wave division multiplexing to provide a peak data rate of over 500 Gbps per fiber. It is packet switched with 2X2 switch nodes. It can deliver a packet to destination location in under 100 nanoseconds, even under heavy loads with a peak bi-section bandwidth of 1 Pbps.

4. Dynamic Adaptive Resource Management

While the global structure of the HTMT system architecture may appear superficially similar to other tightly coupled shared memory multiprocessors, the operation and management of its resource is radically different. Where a conventional system employs a processor or array of processors to both perform the application computation and manage the memory resources, the HTMT architecture employs the smart memory subsystem to manage the computational processor resources. This revolutionary strategy swaps roles of the processors and memory. In a sense, for conventional systems, processors are smart and memory is dumb. In HTMT, the memory is smart and the processors are dumb. In a conventional system, the processors are the masters and the memory the slaves. In HTMT, this is reversed such that the smart memory serves as system master and the computational processors are the slaves. The motivation for this is to maximize efficiency of the computational processors by eliminating the overheads and latency waiting times experienced by the processors. Several mechanisms are devised to accomplish this.

4.1 Multithreaded Computational Processors

Ultra-scale systems will exhibit potentially very large latencies. This is particularly true for systems incorporating components with widely disparate cycle times. HTMT will employ processors with internal clock cycle rates of 100 GHz using superconductor logic while its bulk holographic memory modules will have cycle times on the order of 10 microseconds imposing a worst case latency of a million processor cycles. Even to DRAM, latencies on the order of 10,000 cycles are possible and to the high speed SRAM a latency of several hundred cycles is possible.

HTMT addresses part of the challenge of latency through the use of advanced multithreaded processor architecture. Multithreading allows the critical resources of a processor to share work from multiple instruction streams simultaneously. Each instruction stream or thread operates under separate program counter control. When an instruction is issued from one thread, the operation may take some number of cycles to complete. In its most simple form, a thread will delay issuing its next instruction until its previous instruction has completed. In the meantime, other threads issue their next instructions with control passing from thread to thread in a single cycle. When a thread makes a remote access to high speed buffer or main memory, it may be delayed for hundreds or thousands of cycles. But only that thread is suspended, the other threads and the processor execution resources continue to perform useful work, thus providing high efficiency. The HTMT system employing the SPELL architecture developed at SUNY Stonybrook and reflecting advanced concepts developed at University of Delaware employs two levels of multithreading. Within a typical instruction stream of a regular thread there often exists fine grain instruction level parallelism that can between one to a few operations in length and be concurrent with other operations of the same thread. In conventional processors, compilers use this fine grain parallelism through reordering to improve execution pipeline operation. More advanced system use this level of parallelism through the Tomasulo operation scheduling algorithm to provide out or order completion of operations incurring different execution times. Very large instruction word or VLIW processors use this parallelism to organize multiple operations to be performed simultaneously through compile time scheduling. HTMT provides a dynamic means of exploiting such parallelism through ultra-fine grain threads called "strands" that share the same register context. Even within a thread, one strand may be stalled waiting for one of its operations such as a data fetch to complete but other parallel strands can continue to issue their own operations.

Multithreading can be very efficient. Hardware support, such as that demonstrated by the Tera MTA system can eliminate time overhead of context switching. It dynamically adapts to system usage. For example, a memory access may take different number of cycles depending on contention of the shared communication resources as well as bank conflicts. But the threads are self-synchronizing. The exact number of cycles to complete an operation is unimportant. As long as there are enough independent threads to continuously fill in the gaps left by the others, the processor resources will be heavily employed and high efficiency can be attained. For HTMT, the latencies experienced in all operations within the processor array including accesses to its high speed buffer memory and SRAM can be managed effectively by this two levels of multithreading. However, hiding the latencies to main memory would require as much as a thousand times as many threads. The hardware cost of supporting the state of that many threads would be prohibitive. Managing the system latency of a machine like HTMT cannot be achieved through processor multithreading alone, although it can play an important role.

4.2 In-Memory Data-Oriented Operations

Conventional computers rely on spatial and temporal locality of data access to make effective use of caches to overcome the latency times of main memory. Unfortunately, for many applications a significant number of data accesses may not exhibit locality properties necessary for good cache behavior. Data oriented computation in particularly may have many instances where a data element is touched once during a phase of the computation, although it may be accessed again in some later phase. Large irregular data structures often involve manipulations for which access to many of their elements is brief. For large systems, it can take longer to move the data between the memory output and the processor registers than the actual on-chip memory access time. With the advent of processor-in-memory technology, it is possible to perform local operations on data within the memory itself. This is particularly valuable for the case when the data is examined once, perhaps with some simple modification made. Many streaming applications fall in to this category. But the challenge of manipulating irregular data structures requires the processing of pointers that are virtual addresses. Thus, the memory chips have to be able to perform the address translation function that is ordinarily accomplished by the main processors. HTMT has developed an advanced PIM architecture that supports in-situ address translation. This architecture is being devised at the University of Notre Dame and the NASA Jet Propulsion Laboratory. It benefits from work performed by the DIVA project led by USC ISI.

HTMT uses its PIM based main memory to perform some critical computations directly in the memory to reduce the latency delays and overheads required to move such data to the computational processors. Large complex data manipulation tasks can be performed within the main memory incorporating PIM technology. Even operations that involve data distributed across a number of PIM chips can be performed without the intervention of the main computation processors. Matrix transpose, tree traversal, and parallel prefix operations are some examples. Gather-scatter type operations can also be conducted by the main memory itself. Memory management including page migration is another class of overhead operation that can be performed by the memory itself, removing this responsibility from the main processors and thereby improving their effectiveness.

4.3 Percolation Task Scheduling

While multithreading is capable of hiding processor latency to the SRAM high-speed buffer memory, as previously indicated, it is insufficient alone to effectively manage the latency to main memory. A major advance of the HTMT project is the development of the *Percolation,* a method of managing latency by prestaging work and data near the computation processors prior to their accessing this information. By ensuring that all information to be referenced by the processors is previously loaded in fast memory, the long latency delays that would otherwise occur are avoided. Percolation provides a dynamic and adaptive means of accomplishing the same function previously attempted on conventional systems through prefetching. It borrows heavily from the previous body of work of coarse-grained dataflow. Percolation is performed by the PIM main and buffer memories. The critical concept is that the conditions to be satisfied for which a function is to be instantiated are monitored by the PIM-based main memory. These precedent constraints usually include the calculation of the argument or operand values as well as completion of some functions that perform global side effects. The main memory maintains additional control data structures that allow the status of the precedent conditions to be tracked, and when fully satisfied to begin the second phase of percolation. This second phase involves moving all information related to the execution of the new function to the high-speed buffer memory. This may include performing a gather on global data and placing that in the buffer memory as well. To move this data, both levels of the memory hierarchy have to negotiate the transfer and must involve some memory management. But the computational processors are never involved in these overhead functions. The PIM-SRAM layer supports the transfer of the data from main memory. Once it has all arrived, the third phase can begin.

The next part of Percolation is the scheduling of the workload on to the computation processors. The PIM-SRAM layer loads the threads of the function into the thread register banks and the instructions in the local instruction buffers. The computation processors then begin to perform the tasks of these new threads until completion. When the function has finished, the SRAM moves the results back to main memory including the synchronization information for future functions to be performed. At no time are the computation processors involved in any of this data management overhead and almost always are the data references satisfied by access to the high-speed buffers. Thus, very high efficiency of the computation processors can be achieved by the combination of multithreading and percolation.

5. CONCLUSIONS

The HTMT system architecture project is exploring the combination of advanced technologies and dynamic adaptive methods of resource management to provide effective means of realizing Petaflops scale computing. Costs in parts complexity, size, and power are reduced by employing alternative technologies capable of ten to a hundred times improvement in functionality compared to conventional means. Dynamic adaptive methods of resource management are employed to achieve high efficiency and sustained performance of the critical computation processors. Processor-in-Memory technology enables local smart actions to be performed within the memory and off-loading much of the overhead ordinarily performed by the main processors themselves. Percolation is a strategy that allows the memory system to control the computation processors and manage the system latency by prestaging the work near the processors.

Much work remains in the development of the HTMT architecture. A semiconductor version of the architecture is being planned to test many of the advanced concepts without incurring the additional risks of the innovative technologies. This will also allow early development of the system software necessary to manage the resources and computation. Even if any one of the technologies ultimately proves to be infeasible, other technologies can be employed to take its place with a different set of properties. But the general system architecture and the dynamic adaptive resource management strategy will remain the same. Together they present the opportunity for a new class of ultra-scale high performance computers in the 21st century.