
accenture
Strategy | Digital | Technology | Operations
High performance. Delivered.
Hyperscale Computing
Navigating the Hardware Landscape

Why hyperscale?

After at least a decade out of the spotlight, the computing hardware world has once again become a hotbed of new development—inspired by soaring demand for bigger, faster and lower-cost data centers. Indeed, as noted in Accenture's 2014 Technology Vision, hardware is becoming a crucial consideration for businesses striving to "go digital."

After at least a decade out of the spotlight, the computing hardware world has once again become a hotbed of new development—inspired by soaring demand for bigger, faster and lower-cost data centers. Indeed, as noted in Accenture's 2014 Technology Vision, hardware is becoming a crucial consideration for businesses striving to "go digital."

Innovations in hardware such as low-power central processing units (CPUs), graphics processing units (GPUs), non-volatile memory, solid-state storage and optical network interconnects are delivering orders-of-magnitude performance improvements.

Put another way, these technologies are fueling advances in hyperscale computing. By "hyperscale," we mean the physical infrastructure of large, centralized systems that support data centers and enable companies like Google, Inc., Amazon.com, Inc. and Facebook, Inc. to manage vast volumes, variety and velocity of data. We also mean the ability of computer system architectures to scale up as needed to meet increased demand.

Hyperscale computing offers a wealth of opportunities—such as the ability to reach global scale and process real-time data. And since such abilities can set a business apart from its competitors, numerous companies are considering building their own hyperscale hardware architecture and making big data management a core, in-house competency. Many of the newer hardware technologies have entered the market only in the last few years. Thus, they may seem foreign to executives who are accustomed to architecting or managing systems that were state of the art just a few years ago. This can pose a challenge for those wondering what hardware architectures would best help their companies seize the opportunities that hyperscale offers.

This report aims to help by providing the information executives need to weigh in order to make prudent decisions about the future of their company's hardware platforms. Such information can be useful whether decision makers are mulling over using public-cloud infrastructure or building hyperscale capabilities on their organization's premises.

Key trends

Meeting the unprecedented demand for computing systems that can handle big data quickly calls for massive amounts of storage and processing capabilities. Moreover, the need for context-aware sensing and decision making on the part of computers has pushed processing power and decision-making capacity to edge devices such as autonomous cars and smart thermostats.
This same need has also led to increasingly centralized data storage and analytics in the cloud. At the same time, thanks to Moore's Law and economies of scale, computing and data-storage costs have plummeted.

Decreases in processing, memory, storage and networking costs have enabled computer-system vendors and data-center operators alike to build hyperscale systems comprising millions of servers, petabytes of memory, exabytes of storage and networking speeds that reach hundreds of gigabytes per second.

Increased use of the cloud has made the hyperscale picture even more alluring. Thanks to the cloud, computing services are always on and always available—from anywhere—at the scale users need. A data center is the nexus from which all cloud services flow. Many such centers are housed in nondescript, warehouse-sized buildings that reveal nothing of the activity humming inside. Amid whirring fans and refrigerator-sized computer racks is a tapestry of electrical cables and fiber optics weaving everything together—the data-center network.

Data centers aggregate the foundational components of hyperscale computing systems. At the highest level, each data center might be part of a larger cloud architecture. At the lowest level, each server in a data center contains technologies that are undergoing notable transitions—from memory and processors, to storage and networking. In the rest of this report, we examine each of these and more, along with the considerations executives might keep in mind when making hardware decisions.

Computing architectures: Parallel and distributed

Parallel and distributed computing architectures play a key role in hyperscale systems. **Parallel processing** is the simultaneous use of more than one CPU (processor) to execute a program. **Distributed processing** refers to *local-area networks* (LANs) designed so a single program can run simultaneously at various sites.

Parallel processing: SIMD versus MIMD architectures

General-purpose parallel processors fall into two categories:

Single instruction, multiple data (SIMD)

A central controller broadcasts the same instruction to different processors. Each processor then executes the same instruction on its data. SIMD allows for the finest "grain" of parallelism, at the level of instructions (such as ADD or MULTIPLY). (Grain size refers to the number of instructions executed in a processor before synchronizing, or communicating data, with another processor.) However, coordinating parallelism at the finest grain level entails extensive "overhead." Thus, operators don't get the anticipated speed except for some applications, such as image-processing graphics. GPUs made by the company Nvidia Corporation are current versions of SIMD architecture, and the compute unified device architecture (CUDA) programming language allows for SIMD parallelism in GPUs.[1]

Multiple instruction, multiple data (MIMD)

The processors execute different instructions on different data, entailing multiple instruction streams instead of the single instruction stream characterizing SIMD. These machines exploit grain parallelism. However, the tradeoff is the right grain of parallelism (hundreds or thousands of instructions between coordination, which minimizes the overhead). The Cray XC-40, SGI UV and IBM EC-12 are MIMD machines.

Shared memory versus distributed memory in MIMD processors

General-purpose MIMD parallel processors can be further broken down into two categories:

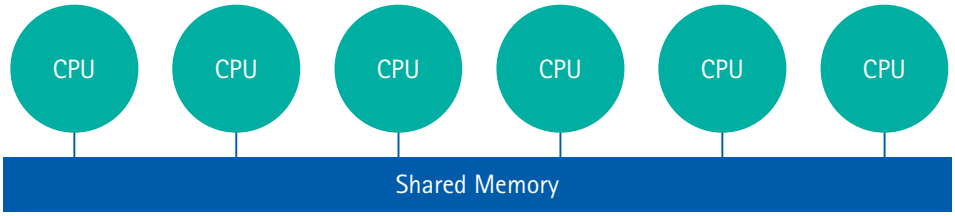
Shared memory

Shared memory parallel processors can access (through Read and Write operations) all memory locations (although at different memory latencies). Shared memory is relatively easy to program but much harder to scale to many processors. Examples of shared memory multiprocessors are the IBM EC12 and the SGI UV systems.

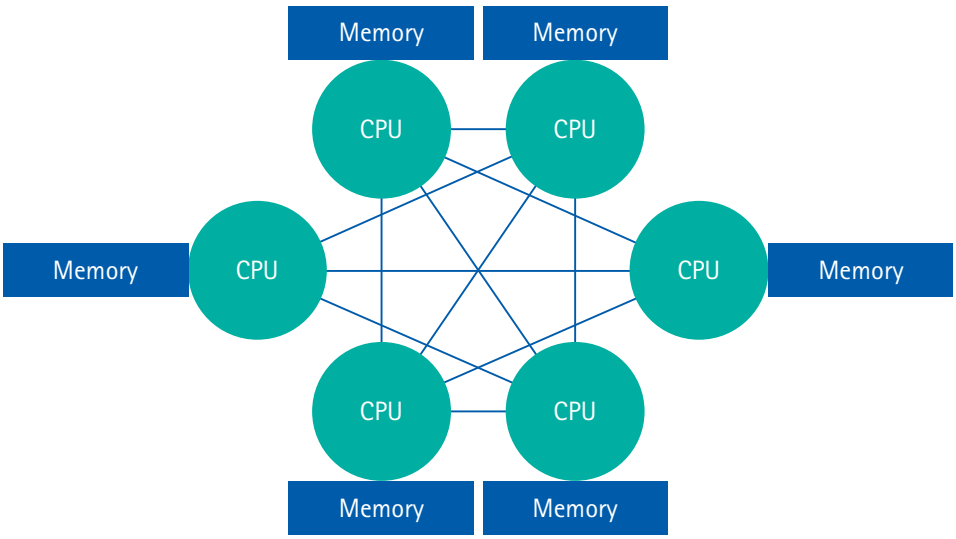
Distributed memory

Each processor has its own local memory to which it can Read or Write. Processors communicate with one another using message Send and Receive protocols (see Figure 1). Distributed memory is easy to scale to thousands of processors but difficult to program, except for very parallel, easy applications. Examples of distributed memory MIMD multiprocessors are the Cray XJ7 and XC40, and the IBM BlueGene/Q systems.

Figure 1: Shared Memory MMID and Distributed Memory MMID Multiprocessor.



(a) Shared Memory MIMD Multiprocessor



(b) Distributed Memory MIMD Multiprocessor

With both shared- and distributed-memory parallel processing, processors are connected through an **interconnection network**. Such networks can take numerous forms, including bus, cross-bar, multistage, rings, 2D and 3D mesh, trees and hypercubes.

The type of interconnection network used depends on the grain size the architecture needs to support (see Table 1). Fine-grain processing is generally used for applications that have extensive inherent parallelism at the finest level, such as image processing or graphics applications that add 1 to each pixel, or that average all pixels in a 1000x1000 image. Coarse-grain processing is more useful in more complex applications, such as matrix operations or data search.

Table 1: How grain size differs.

Grain size	Processors synchronize at...
Fine	Every tens of instructions
Medium	Every hundreds of instructions
Coarse	Every tens of thousands of instructions

A closer look at hyperscale computing systems

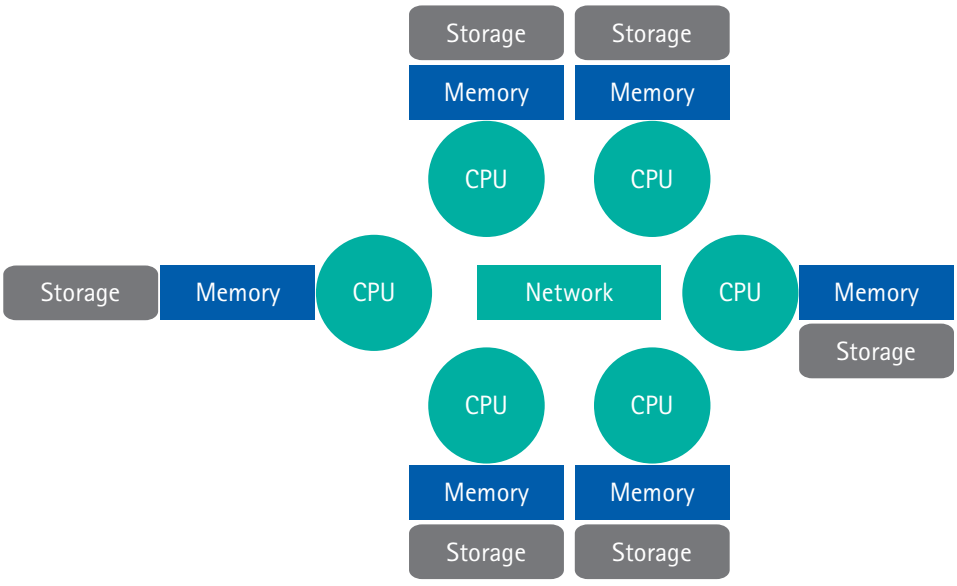
In the world of computing, the word hyperscale refers to an architecture's ability to scale as needed to meet increased demand on the computing system. Scaling typically involves seamlessly provisioning and adding compute, memory, networking and storage resources to a given node or set of nodes that make up a larger distributed or grid computing environment. Hyperscale computing is essential for building a robust and scalable cloud, big data, MapReduce or distributed storage system.

Hyperscale computing systems used by companies like Google, Amazon and Facebook are, in essence, giant distributed-memory MIMD systems. The processors are connected to their own local memories. But each processor (or server) also has its own disk for storing its non-volatile data, or data that remains after power events or machine reboot cycles.

Hyperscale systems can be described as "scale-out." Such systems are characterized by modular designs made of server units, or nodes (each of which comprises a processor, memory and storage), that are replicated using loosely connected interconnection networks and parallel programming. These systems can be scaled linearly to millions of servers, each with multicore processors, hundreds of gigabytes of random access memory (RAM) and terabytes of disk space. Collectively, such a system will have millions of processors, petabytes of RAM and exabytes of storage (see Figure 2).

Figure 2: Hyperscale system.

Each node consists of processors, memory and storage that is replicated at scale and connected through commodity interconnects.



Open Compute Project

In April 2011, Facebook announced the Open Compute Project initiative, which aimed to foster the sharing of data center product designs.[2] The initiative's end goal was to accelerate the design of hardware systems—similar to the Open Source movement in the software world. The designs aim to improve energy efficiency. In addition to reducing energy consumption and cost, Open Compute increases reliability and choice in the marketplace, and simplifies operations and maintenance. The project started with the opening of the specifications and mechanical designs for major data-center components, which include the following:

Server compute nodes

Open Compute motherboards are power-optimized, barebones designs that provide the lowest capital and operating costs. There is one for Intel processors, one for AMD processors and one for ARM processors.

Storage nodes

Storage is a key component of any data center, and offers many opportunities for efficiency gains. There is a specification for cold data storage nodes, where data is stored on disk but almost never read again. There is a second specification for the Open Vault, which is a simple and cost-effective storage solution that offers high-disk densities, holding 30 drives in a 2U chassis. There is third specification for a solid-state disk (SSD) storage using Fusion-io 3.2 terabytes as a high-density I/O PCI Express adapter card.

Open networking

The Open Compute Networking Project is creating a set of technologies that are disaggregated and fully open, allowing for rapid innovation in the network space. The objective is to disaggregate networking hardware from networking software using concepts such as software-defined networking. Two switch designs (one by Mellanox Technologies and the other by Broadcom Corporation) have been designed.

Data center rack

The data center maximizes mechanical performance and thermal and electrical efficiency. It accepts 277 volts of AC, so more energy makes it from the grid to the data center to server components.

Any user of hyperscale systems can take a look at the open computing standards for compute, storage, networking and data centers, and work with solution providers such as Hyve Solutions, AMAX Information Technologies, Inc., Quanta Computer or Penguin Computing to have hyperscale systems custom designed to specifications. This is a real alternative to traditional system vendors such as Hewlett Packard Development Company L.P., Dell, Inc. and IBM.[3]

Questions for your next meeting

How familiar are you and your peers with the new hardware technology that supports hyperscale computing?

What competitive advantages could hyperscale computing give your organization?

What do you see as your organization's biggest challenges related to making decisions about hyperscale hardware architecture?

What kinds of computing architectures—parallel and distributed, SIMD and MIMD, interconnection networks—is your organization currently using? What do you see as their benefits and difficulties? How might you get more value from them?

Are you considering Open Compute Project to design your hyperscale data centers?

High-performance computing systems

Hyperscale systems are "scale-out" and made of nodes replicated using a network and parallel programming. By contrast, high-performance computing systems can be described as "scale-up." In such systems, the processors and memories are tightly integrated with a fast network, and users seek to speed up a particular application, such as weather modeling or computational fluid dynamics.

High-performance computing systems are also known as supercomputers, and constitute the most powerful computer systems in the world at a given time in terms of computing power, memory storage and disk storage. The components (CPUs, memory, storage, interconnects, operating system) are carefully integrated to provide the highest performance in a single application that can be executed in parallel.

There are two categories of high-performance computing systems. The first is based on shared Memory MIMD architectures. An example of a state-of-the-art shared memory MIMD multiprocessor is the Silicon Graphics® SGI UV™ product, which consists of 2048 cores (4096 threads) using the latest Intel® Xeon® processor E5-4600 product family.[4] The SGI UV system provides a scalable, coherent shared memory system because of its innovative NUMAlink® interconnect. The largest UV system provides up to 64 terabytes of main memory. SGI also supports graphics accelerator cards using Nvidia® Quadro®, Nvidia® Tesla® K20 GPU computing accelerator and Intel® Xeon Phi™.

These machines provide a shared memory programming model to application developers for data-intensive applications; they are much easier to program but very hard to scale.

The second is based on distributed memory MIMD architectures. An example of a state-of-the-art distributed memory MIMD multiprocessor is the Cray XC40™ system, which uses the custom Aries interconnect, Intel® Xeon® processors, integrated storage solutions, and major enhancements to the Cray OS and programming environment.[5] The processors used are the Intel Xeon processors. There is an option to include integrated graphics processing units (GPUs) coprocessors using the Intel Xeon Phi or the Nvidia Tesla K40 GPUs. The Cray XC series supercomputer can deliver up to 100 petaflops per system. These machines are harder to program but easier to scale. The highest performing supercomputers in the world are all based out of distributed memory MIMD architectures and all use the message passing interface (MPI) programming model.

TOP 500 supercomputers

Twice a year, the TOP500 project ranks and describes the 500 most powerful computer systems in the world.[6] It aims to help decision makers track and detect trends in high-performance computing (see Table 2).

Table 2: List of Top Five Supercomputers announced November 18, 2014.

Computers are ranked first by performance. There are two performance measurements. The first is the highest sustained score measured using the LINPACK benchmark suite, measured in petaflops. The second is the computer's theoretical peak performance, measured in petaflops.

Rank	Name	Installation	Vendor	Processors, interconnect	Performance	Power	Memory
1	Tianhe2	National Supercomputer Center Guangzhou	NUDT	3.1 million cores, Intel Xeon E5-2692v2 12C 2.2 gigahertz (GHz) processor, custom interconnect	**33.9 petaflops sustained (pf sust)**, 54.0 petaflops peak (pf peak)	17.8 megawatts (MW)	1024 terabytes (TB)
2	Titan XJ7	Oak Ridge National Lab	Cray	560, 640 cores, AMD Opteron 6274 16C 2.2GHz, Gemini interconnect	**17.6 pf sust**, 27.1 pf peak	8.2 MW	710 TB
3	Sequoia Blue Gene/Q	Lawrence Livermore National Lab	IBM	1,572,864 cores, Power BQC 16C 1.6GHz processor, custom interconnect	**17.1 pf sust**, 20.1 pf peak	7.9 MW	1,572 TB
4	K Computer	RIKEN Advanced Institute of Computational Science	Fujitsu	705,024 cores, SPARC64 VIIIfx 8C 2GHz processor, custom interconnect	**10.5 pf sust**, 11.3 pf peak	12.7 MW	1,410 TB
5	Mira Blue Gene/Q	Argonne National Laboratory	IBM	786,432 cores, Power BQC 16C 1.6GHz processor, custom interconnect	**8.6 pf sust**, 10.0 pf peak	3.9 MW	1,572 TB

Figure 3 shows the relative breakdown of the top 500 supercomputers by vendors. While the top 10 were developed by IBM (5), Cray (2), Dell (1), Fujitsu (1) and NUDT (1), the distribution of vendors changed in the top 500 with HP (179), IBM (153) and Cray (63) being the top three vendors.

Table 3 shows the progression of the most powerful computer in the world (number 1 ranked computer) in the top 500 list from 1993 to 2014. The performance has increased six orders of magnitude (nearly a million times over past 20 years).

The top 500 supercomputers are overwhelmingly based on x86-64 (the 64-bit version of the x86 instruction set) CPUs.

Figure 3: Breakdown of top500 supercomputers by vendors.

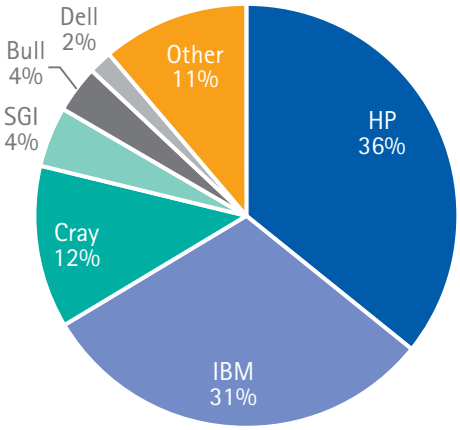


Table 3: Performance of most powerful supercomputer in the world during 1993-2014.

Year	Performance as measured by sustained LINPACK performance
1993	59.7 GigaFlops
1994	143.4 GigaFlops
1995	170 GigaFlops
1996	220.4 GigaFlops
1997	1.1 TeraFlops
1998	1.3 TeraFlops
1999	2.1 TeraFlops
2000	2.4 TeraFlops
2001	7.2 TeraFlops
2002	35.9 TeraFlops
2003	35.9 TeraFlops
2004	35.9 TeraFlops
2005	136.8 TeraFlops
2006	280.6 TeraFlops
2007	280.6 TeraFlops
2008	1.0 PetaFlops
2009	1.1. PetaFlops
2010	1.8 PetaFlops
2011	8.2 PetaFlops
2012	16.3 PetaFlops
2013	33.9 PetaFlops
2014	33.9 PetaFlops

Supercomputer power consumption and efficiency

High-performance computing systems consume a lot of power. Most of the power used is converted into heat, so a system that needs fewer watts to do a job will require less cooling to maintain a given operating temperature. Lower energy consumption can also make the computer less costly to run, and can reduce the environmental impact of powering the computer. If a lower-power computer is installed in a location with limited climate control, it will operate at a lower temperature, which may make it more reliable. In a climate-controlled environment, reductions in direct power use may also deliver climate-control energy savings.

Common power-consumption metrics include FLOPS (floating point operations per second) and MIPS (million instructions per second). A recent metric used to compare supercomputers is FLOPS per watt (). Power usage efficiency (PUE) also matters in high-performance (as well as hyperscale) computing systems. PUE is the ratio of total amount of energy used by a computer data center (including energy for lighting and cooling) to the amount of energy delivered to the computing equipment. A PUE of 1.0 is the ideal (implying there is no overhead). Typical computer centers have PUEs of 1.1 to 1.2. Architects of hyperscale data centers strive for PUEs close to 1.0.

Exascale Computing Systems

The fastest supercomputer in the world in 2014 delivers a sustained computing capacity of 34 PetaFLOPS with a total system memory of 1 petabyte at an energy consumption of 18 megawatts. The Department of Energy (DOE) has recently announced the Exascale Computing Systems Challenge to challenge the research community and the supercomputing vendors to design and build an exascale computing system capable of maintaining sustained 1 ExaFLOP computing performance (1,000 PetaFLOPS) and 10 petabytes of memory with a total power consumption not to exceed 20 megawatts.[7] An exaflop (10 ^ 18 FLOPS = 1,000 petaFLOPS) system would need a R&D effort to span a wide range of areas, including hardware, systems software, application algorithms and computer science.

DOE has been funding research contracts at supercomputer vendors to design such an exascale computing system. On November 14, 2014, DOE announced the winners of three such systems all rated to perform at approximately 150 Petaflops, two of which will be designed by IBM, and one of which will be designed together by Intel and Cray. The IBM systems will be installed at Lawrence Levermore and Oak Ridge National Labs, and the Intel/Cray system will be installed at Argonne National Labs.

The IBM systems will be rated to perform at 150 PetaFlops and will be based on the new IBM OpenPower-based processor architectures with more than five petabytes of dynamic and flash memory to help accelerate the performance of data-centric applications. The Intel/Cray system will also be rated at 150 PetaFLOPS and will be based on the Intel Xeon Phi processors and the Aries interconnect.

Questions for your next meeting

In what respects does your organization use or benefit from high-performance computing systems, or supercomputers?

Which computing-system ability—"scaling out" or "scaling up"—is most important for your organization?

Regarding the computing systems your organization currently uses, what opportunities and challenges are you experiencing related to power consumption and efficiency?

Parallel and distributed programming: Basic models

To understand how hyperscale and high-performance computing systems are programmed, it's helpful to know about some basic programming models. Understanding the differences between data and functional parallelism, seeing examples of SIMD and MIMD programming, and learning more about MapReduce programming are good places to start.

Data versus functional parallelism

There are two common forms of parallel programming:

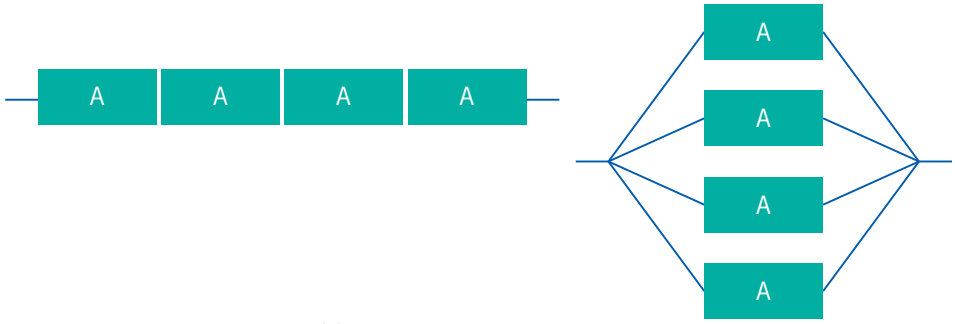
Data parallelism

This form involves creating multiple identical processes and assigning a portion of the data to each process. It is appropriate for applications that perform the same operations repeatedly in large collections of data, and for applications requiring loops to perform calculations on arrays or matrices. In such applications, loop iterations are executed in parallel.

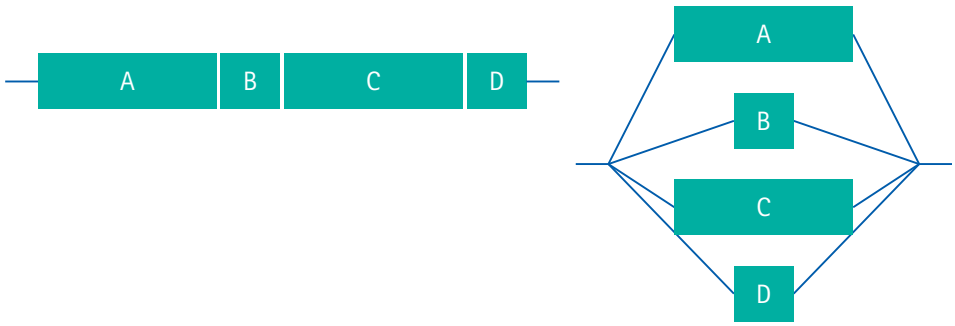
Functional parallelism

Also called task parallelism or control parallelism, functional parallelism involves creating multiple processes and having them perform different operations on a shared data set. It is appropriate for applications that include many unique subroutines, procedures or functions, each of which can be executed in parallel (see Figure 4).

Figure 4: Examples of data and functional parallelism.



(a) Example of Data Parallelism



(b) Example of Functional Parallelism

MIMD and SIMD parallel programming examples

Consider the following excerpt from a C program:

```
int a[100] b[100] c[100] /* declare
arrays a, b, c of 100 elements */

main ()

{

        for (i=0; i < 100; i++)

                a[i] = b[i+1]+ b[i] * c[i];

}
```

Below is the shared memory MIMD version of the code excerpt. Tasks are scheduled by the processors at runtime. For a loop having N iterations on P processors, processor 1 would be allocated iterations 0, N/P-1, and processor 2 would be allocated N/P to 2N/P-1. Assuming five processors, each processor would execute 20 iterations of the loop.

```
int a[100] b[100] c[100] /* declare
arrays a, b, c of 100 elements */

main ()

nprocs = get_num_procs(); /* number of
processes */

id = fork(nprocs); /* create nprocs
processes */

lb = id*100/nprocs; /* the lower bound
of chunk of iterations */

ub = (id+1)*100/nprocs; /* the upper
bound of chunk of iterations */

{

        for (i=lb; i < ub; i++)

                a[i] = b[i+1]+ b[i] * c[i];

}
```

An excellent overview of shared memory parallel programming appears in [REFERENCE Bauer 1989] and [REFERENCE Brawer 1989]. In the past, each system vendor (SGI, HP, IBM and others) used to provide their own custom form of shared memory programming using process creation, and synchronization using locks and unlocks. The POSIX threads library (called Pthreads) has been developed as a portable threads library for shared memory MIMD programming. While this explicit thread-style programming is cumbersome to program, several directive-style programming methods have been created. A programmer programs using easy-to-use compiler directives and the compiler generates the lower level calls to shared memory constructs like Pthreads libraries. The OpenMP consortium has been created to standardize directive-based, multilanguage, high-level parallelism that is performant, productive and portable.[8]

Below is the distributed memory MIMD version of the same code excerpt. Let's assume that there are four processors in a linear array: 0, 1, 2, 3. By distributing the arrays a, b and c of 100 elements among the four processors, each processor would have 25 elements of a, b and c. For each processor p, the array element b[25] that is needed for iteration 24 must be obtained from processor p+1. So, the array declaration of b is extended by 1 to b[26] instead of b[25].

```
int a[25] b[26] c[25] /* declare arrays a,
b, c of 25 elements */

main ()

{

        id = mynode();

        send(BTYPE, &b[0], 4, id+1,0); /*
send one element on the boundary to
the neighboring process */

        receive(BTYPE, &b[25], 4); /*
receive one element on the boundary
from neighboring process */

        for (i=0; i < 25; i++)

                a[i] = b[i+1]+ b[i] * c[i];

}
```

The message passing interface (MPI) is a standard that has evolved over the years to support message passing programming in distributed memory MIMD multiprocessors. An excellent overview of writing applications on message passing MIMD machines is in [REFERENCE G.C. Fox 1984]. All vendors of large-scale MIMD systems support MPI. The Open MPI Project [REFERENCE OpenMPI] is an open source message passing interface implementation that is developed and maintained by a consortium of academic, research and industry partners.

Finally, here is a SIMD version of the code excerpt. The parallel function a = b2 + b * c means that each array element will equal the sum of the left-shifted version of b and the product of the elements b and c.

```
shape [100] s;

int: s a, b, c /* declare arrays a, b, c of
shape s of 100 elements */

main ()

                b2 = left-shift(b); /*
shift elements of b to the left by one
element */

                a = b2 + b * c;

}
```

CUDA® is a parallel computing programming model developed by Nvidia that leverages SIMD data parallelism. It enables dramatic increases in computing performance by harnessing the power of the GPU.[9]

CUDA is specifically for Nvidia GPUs while OpenCL is designed to work across a multitude of architectures, including GPU, CPU and DSP (using vendor-specific software developer kits (SDKs).

MapReduce Programming

The MapReduce programming model has recently become popular for big data applications executed on hyperscale computing systems. MapReduce programming is used for processing and generating large data sets with a parallel, distributed algorithm on a **cluster**—a collection of nodes.

A MapReduce program comprises two key procedures:

- **Map()** performs filtering and sorting.
- **Reduce()** performs a summary operation.

The MapReduce system marshals distributed servers, runs tasks in parallel, manages all communications and data transfers among the system's parts, and provides for redundancy and fault tolerance.

Questions for your next meeting

Is data parallelism or functional parallelism most appropriate for the applications your organization uses?

Will your applications be more suitable for shared memory MIMD parallelism or message passing interface distributed memory MIMD parallelism?

Can your applications leverage SIMD-style fine-grain data parallelism? Have you considered GPUs such as Nvidia processors?

Does your organization use the MapReduce programming model? If so, what advantages does the model offer? What challenges does it present? How might you get more value from this model?

Processor architectures

To understand the processor architectures used in hyperscale and high-performance computing systems, it's helpful to know something about instruction set architectures, multicore processors, Intel and ARM processors, GPUs and CPU/GPU combinations.

Processor instruction set architectures

A computer's CPU reads machine instructions from main memory and executes them. Here are two examples of instructions:

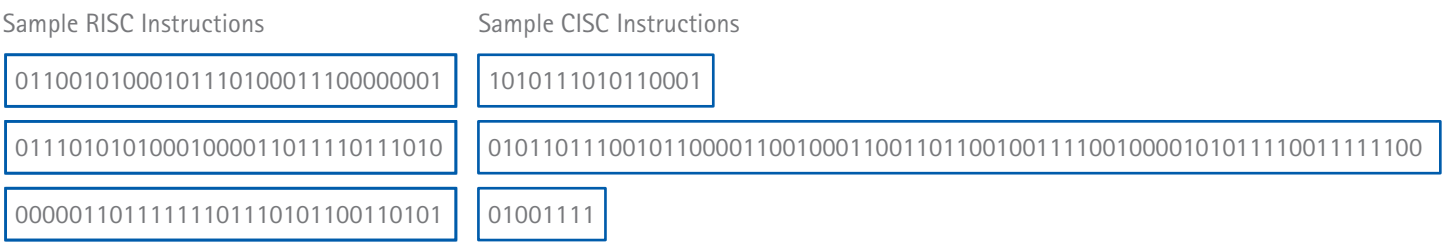
- ADD R1, R2, R3. This instruction reads the contents of register R1 and register R2, adds them and stores them in register R3. (A register is a fast local storage area on a CPU.)
- LOAD R1, memory(R2). This instruction loads the contents of main memory pointed to by register R2 into register R1.

A CPU's instruction set architecture (ISA) consists of the specification of the types of instructions that a processor can execute and the format of the instructions. There are two types of processor ISAs:

- CISC: complex instruction set computer
- RISC: reduced instruction set computer.

The primary difference between CISC and RISC is the binary length of the instructions (such as ADD, MOVE DATA and BRANCH) that each type can handle. RISC processors accommodate a fixed-length instruction (generally, one instruction per clock cycle). CISC can handle instructions of variable length and may take several clock cycles to execute a single instruction (see Figure 5).

Figure 5: RISC vs. CISC instruction sets.



Each ISA type has pros and cons. For instance, CISC requires much less complex compilers than RISC does, makes more efficient use of memory and shifts the burden of generating machine instructions to the processor. However, CISC CPUs also consume much more energy than their RISC counterparts.

RISC processors, on the other hand, are generally less complex and shift more of the complexity burden to software instead of hardware. This minimizes a program's runtime complexity, improving runtime performance.

Over time, as clock speeds and transistor density increased, manufacturers have moved to ISAs with multiple cores. Some of the best advantages of each ISA type became adopted in the other. However, use of any type of CISC processor where power consumption is a design constraint (such as compute-intensive mobile devices) is out of the question. Consequently, RISC has begun to dominate.

Multicore Processors

A multicore processor is a single computing component with two or more independent CPUs ("cores"), which read and execute program instructions (). Multiple cores can run multiple instructions at the same time, increasing overall speed for programs amenable to parallel computing. Manufacturers typically integrate the cores on a single integrated circuit die (known as a chip multiprocessor, or CMP), or on multiple dies in a single chip package. A multicore processor implements MIMD multiprocessing in a single physical integrated circuit (IC) chip.

Multicore processors are widely used across many application domains. The improvement in performance gained by the use of a multi-core processor hinges on the software algorithms used and their implementation. For example, gains may be limited by the fraction of the software that can be run in parallel simultaneously on multiple cores.

Many-core and massively multicore architectures have an especially high number of cores. For example, the IBM Power8 processor has 12 cores, the AMD OpteronTM processor has 16 cores, the Intel Xeon Phi processor has 60 cores, the Nvidia GeForce processor has 240 cores and the Tilera TILE64 has 64 cores. To make these, chip manufacturers have had to rely on parallelism to extract more performance from the architectures. Yet it is difficult to extract parallelism out of such architectures for a single application.

Intel, IBM and ARM Processors

Intel has developed a number of generations of processor microarchitectures (which have a high-level instruction set in terms of how the CPU's cores, registers, caches and other components are organized). **Sandy Bridge** was the codename for a microarchitecture developed by Intel beginning in 2005 for CPUs to replace the Nehalem microarchitecture. Intel demonstrated a Sandy Bridge processor in 2009, and released the first products based on the architecture in January 2011 under the Intel Core brand. Sandy Bridge implementations targeted a 32 nanometer manufacturing process. **Ivy Bridge** is the codename for a line of processors based on the 22 nanometer manufacturing process. Ivy Bridge processors are backwards-compatible with the Sandy Bridge platform. **Haswell** is the codename for a processor microarchitecture developed by Intel as the successor to the Ivy Bridge architecture. It uses the 22 nanometer manufacturing process. With Haswell, Intel introduced a low-power processor designed for Ultrabooks. Haswell CPUs are used in conjunction with the Intel 8 Series chipsets and Intel 9 Series chipsets. **Broadwell** is Intel's codename for the 14 nanometer die shrink of its Haswell microarchitecture due in late 2014. Broadwell will be used in conjunction with Intel 9 Series chipsets. **Skylake** is the codename for a processor microarchitecture to be developed by Intel as the successor to the Broadwell architecture. Skylake will use the 14 nanometer manufacturing process. The first Skylake processors will be desktop Core i5 and Core i7 parts. Future hyperscale computing systems using scale-up architectures will be based on the Skylake processors from Intel.

ARM is another notable instruction set architecture (ISA) for computer processors. Developed by ARM Holdings, ARM uses a RISC-based design approach that enables the use of far fewer transistors, which reduces power consumption. Many smartphones, digital televisions and mobile computers use chips based on ARM architectures. While ARM processors were originally developed for low-power applications and devices such as smartphones and tablets, ARM processors are now being designed for servers. While the Intel x86 ISA was manufactured by Intel and AMD only, ARM-based processors are being designed and manufactured by companies such as Calxeda, Cavium, Applied Micro, Broadcomm, Qualcomm, Texas Instruments and Nvidia.

In 2011, HP announced the project Moonshot and showed a prototype Redstone platform, a server based on an ARM server-on-chip by Calxeda. On September 29, 2014, HP announced general availability of two models of ARM servers in its Moonshot line. The ProLiant m400 is a general-purpose machine, built primarily for large-scale cloud service providers and Internet companies, and powered by X-Gene, the 64-bit ARM server-on-chip by Applied Micro. The ProLiant m800 is powered by a 32-bit ARM server-on-chip by Texas Instruments and is more of a niche product, aimed at highly specialized workloads that can take advantage of Texas Instrument's advanced digital signal processing capabilities.

While ARM server chips were initially available only for 32 bits, more recently, 64 bit ARM processors have been announced. This will now start a whole new set of hyperscale server architectures based on the ARM ISA. Hyperscale system vendors such as HP, Dell and IBM now have a lot more choice for processor vendors. Cloud computing companies such as Facebook, Amazon and Google are also considering the use of ARM-based servers in their systems.

The third ISA that is used in current generation high performance servers is the Power Architecture products from IBM. However, these processors were only used in the IBM Power Systems architectures alone, and also in the highest end supercomputers such as the IBM Blue Gene system. On August 6, 2013, a group of companies (Google, IBM, Mellanox, Nvidia and Tyan Computer Technology Corporation) announced plans to form the OpenPower Consortium—an open development alliance based on IBM's Power microprocessor architecture. The Consortium (which now includes companies such as Altera Corporation, SK Hynix, Inc., Fusion-io and Micron Technology, Inc.) intends to build advanced server, networking, storage and GPU-acceleration technology aimed at delivering more choice, control and flexibility to developers of next-generation, hyperscale and cloud data centers.

Graphics processing units

GPUs are chips designed to rapidly manipulate and alter memory to accelerate the creation of images for output to a display. They have a highly parallel structure and can usually be replaced or upgraded relatively easily, assuming the motherboard can support the upgrade. Dedicated GPUs for portable computers are most commonly interfaced through a non-standard and often proprietary slot, owing to size and weight constraints.

Integrated graphics processors (IGPs) use a portion of a computer's system RAM rather than dedicated graphics memory (). IGPs can be integrated on the motherboard as part of the chipset, or in the same die as the CPU.

However, a GPU is extremely memory intensive, so an integrated solution may compete with the CPU for the already relatively slow system memory (RAM). Moreover, today's high-performance GPUs may be the largest power consumers in a hyperscale or high-performance computing system. Peak performance of any system is limited by the amount of power the system can draw and the amount of heat it can dissipate.

Increasingly general-purpose GPUs are being used as a modified form of stream processor. In certain applications, this can yield several orders of magnitude higher performance than a conventional CPU.

Nvidia GPU cards support API extensions to the C programming language such as CUDA and OpenCL. CUDA is specifically for Nvidia GPUs while OpenCL is designed to work across a multitude of architectures including GPU, CPU and DSP (using vendor specific SDKs).

CPU/GPU combination

The latest hyperscale servers from Facebook, Google and Amazon are experimenting with combining ARM-based CPUs with GPUs to get the best power performance. More recently, application developers have realized that these GPUs are not just an application-specific integrated circuit (ASIC) for graphics processing. The same architecture can be used for other applications as well, such as image processing, data processing and analytics.

Intel has announced the Xeon Phi processors, which integrate multicore Intel X86 processors with GPUs on a chip. These integrated processors are used in numerous supercomputers. The latest Nvidia GPU architectures are the Tesla K40 GPU. Nvidia has recently announced a partnership with IBM (and other companies such as Google, Mellanox and Tyan) on the OpenPower Consortium where it will be possible for Nvidia GPUs to talk directly to IBM Power architectures using a common platform on a chip.

Today, many supercomputers also combine ARM-based CPUs with GPUs to get maximum performance with minimum power consumption. The fastest supercomputers have 16-32-way parallelism per node, with as many as 1,000 nodes in the largest supercomputer systems.

Questions for your next meeting

Is your organization currently tied to the Intel x86 architecture and, therefore, restricted to the Intel Xeon Phi processors? Are you considering other architecture choices such as the ARM or IBM OpenPower?

Given the resurgence of interest in ARM-based processors, what might be the pros and cons of getting a custom version of ARM for your organization, with your own server-on-chip building blocks (networking, Ethernet)?

What advantages might CPU/GPU combination give your organization? (Examples might include the ability to exploit fine-grain parallelism at the GPUs using SIMD and coarse-grain parallelism with multiple cores using MIMD.) For which applications might you consider ASIC or field-programmable gate array (FPGA) coprocessors?

Memory

Computer systems use different levels of caches to store memory references to most recently used instructions and data:

- Level 1 caches are typically built out of static random access memory (SRAM) and have 1-2 cycle latency (typical size of 32-64 kilobytes).
- Level 2 caches are built out of dynamic random access memory (DRAM) and have 4-6 cycle latency (typical size of 128-256 kilobytes).
- Level 3 caches are built from DRAM and have 6-8 cycle latency (typical size of 2-8 megabytes).
- Main memory external to the processor chip uses DRAM and can have hundreds of clock cycles of latency (typical size of 64-512 gigabytes).
- The largest supercomputers in the world in 2014 have up to 1 petabyte (1 million gigabytes) of DRAM memory.

Below, we take a closer look at SRAM and DRAM.

Static Random Access Memory (SRAM)

Unlike DRAM, SRAM does not have to be periodically refreshed. It exhibits data permanence, but is still volatile in the sense that data is eventually lost when the memory is not powered.

A typical SRAM storage cell comprises metal-oxide semiconductor field-effect transistors (MOSFETs). Each bit in an SRAM is stored on transistors that form cross-coupled inverters This storage cell has two stable states, which are used to denote 0 and 1. Additional transistors control access to a storage cell during Read and Write operations.

Generally, the fewer transistors needed per cell, the smaller each cell can be. Since the cost of processing a silicon wafer is relatively fixed, using smaller cells and (as a result) being able to pack more bits on one wafer reduces memory cost per bit.

Dynamic Random Access Memory (DRAM)

DRAM stores each bit of data in a separate capacitor within an integrated circuit (). The capacitor can be charged or discharged; these two states are taken to represent the two values of a bit, conventionally called 0 and 1. Since even non-conducting transistors always leak a small amount of charge, the capacitors will slowly discharge. The information eventually fades unless the capacitor charge is refreshed periodically.

With DRAM, only one transistor and a capacitor are required per bit. This lets DRAM reach very high densities of memory bits per unit area on a memory chip. On the other hand, DRAM is volatile, since it loses its data quickly when power is removed. Yet the transistors and capacitors used are extremely small—billions can fit on a single memory chip.

Caches

A CPU uses a cache to reduce the average time needed to access memory. When the CPU needs to Read from or Write to a location in the main memory, it first checks whether a copy of that data is in the cache. If so, the CPU immediately Reads from or Writes to the cache, which is much faster than doing so with main memory.

Most modern server CPUs have:

- An **instruction cache** that speeds up executable instruction fetch.
- A **data cache** that speeds up data fetch and store, and that typically is organized as a hierarchy of levels.

Data is transferred between memory and cache in blocks of fixed size, called cache lines. When a cache line is copied from memory into the cache, a cache entry is created. The cache entry will include the copied data as well as the requested memory location (now called a tag).

When the CPU needs to Read from or Write to a location in main memory, it first checks for a corresponding entry in the cache. The cache checks for the contents of the requested memory location in any cache lines that might contain that address. At this point, one of two events could happen:

- **Cache hit:** The CPU finds that the memory location is in the cache, and immediately Reads or Writes the data in the cache line.
- **Cache miss:** The CPU does not find the memory location in the cache. The cache allocates a new entry and copies in data from the main memory. Then the CPU fulfills the request from the cache's contents.

Multilevel caches

Computing systems make a tradeoff between cache latency and hit rate. Larger caches have better hit rates but longer latency. To manage this tradeoff, many computers use multiple cache levels—with small, fast caches backed up by larger, slower caches. With multilevel caches, the CPU generally checks the fastest, *level 1* (L1) cache first. If it "hits," the CPU proceeds at high speed. If that smaller cache misses, the CPU checks the next fastest cache (*level 2*, or L2), and so on, before it checks external memory.

The latency difference between main memory and the fastest cache has become larger because DRAM technology has not improved as quickly as processor IC technology. As a result, some CPUs have been designed to use as many as three levels of on-chip caches.

Caches for multicore chips

When considering a chip with multiple cores, the question arises of whether caches should be shared across cores or local to each core. Cache sharing introduces more wiring and complexity. But having one cache per *chip*, rather than *core*, greatly reduces the amount of space needed; thus the system can include a larger cache.

Typically, sharing the L1 cache between multiple cores is undesirable—because the resulting latency increase causes each core to run considerably slower than a single-core chip. On the other hand, for the highest cache level (the last one the CPU calls before accessing the main memory), having a global cache allows for better data sharing between cores as well as the exploitation of more parallelism.

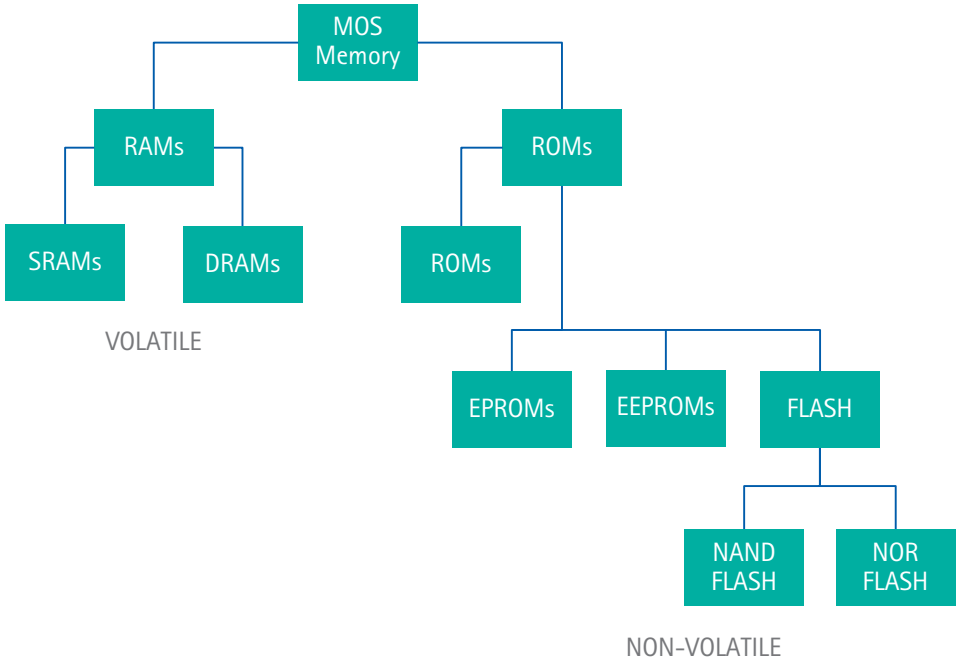
Non-volatile memory

Non-volatile memory retains its information when power is turned off. Such memory can take the form of non-volatile random access memory (NVRAM) as well as various types of non-volatile read-only memory (see Figure 6).

The high power needed to Write the cells is a problem in low-power roles, where NVRAM is often used. A number of new memory devices have been proposed to address these shortcomings:

- **FRAM.** Ferroelectric RAM, or FRAM, contains a thin ferroelectric film in which atoms change polarity in an electric field, producing a binary switch. Unlike RAM devices, FRAM retains its data memory when power is shut off or interrupted. Its distinct properties include extremely high endurance, ultra-low power consumption, single-cycle Write speeds and gamma radiation tolerance.
- **MRAM.** Magneto-resistive RAM, or MRAM, uses magnetic elements and operates like core memory, at least for the first-generation technology. Newer-generation techniques appear to allow for much higher densities than the first generation, but are lagging behind Flash in terms of sales for the same reasons FRAM does: enormous competitive pressure in the Flash market.

Figure 6: Memory taxonomy.



- **PCRAM.** Phase-change RAM, or PCRAM, is based on the same storage mechanism as writable CDs and DVDs, but reads them based on their changes in electrical resistance rather than changes in their optical properties. PCRAM has higher areal density than modern Flash devices.

A number of more esoteric devices have been proposed, including nano-RAM based on carbon nanotube technology. The advantages nanostructures offer over their silicon-based predecessors include their tiny size, speed and density. However, these are currently far from commercialization. Several concepts for molecular-scale memory devices have also been developed recently, and these will merit watching in the near future.

Questions for your next meeting

If you are considering a private data center for your organization, what advantages and challenges might the newer forms of memory present?

What forms of memory do your organization's computing systems currently use most?

What changes (if any) in your organization's current choices of memory technology might be worth considering?

Storage

All computer systems need some form of non-volatile storage so that any data created or written by a computer program persists beyond the life of the application. In this section, we describe technologies developed for hyperscale computing that enable computer systems to store large amounts of data permanently. Such technologies include disk storage, solid-state drives, network-attached storage, storage-area networks, tiered storage, hierarchical storage management and software-defined storage.

Disk storage

With disk storage, data is recorded when electronic, magnetic, optical or mechanical changes are made to a surface layer of one or more rotating disks, which may be made of a variety of materials. The disk drive stores data onto cylinders, heads and sectors. A sector represents the smallest size of data to be stored in a hard disk drive. Each file—or logical organization of data on a disk for sequential access—will have many sectors assigned to it.

Data is stored through changes in the physical properties (optically or magnetically, for example) of each byte on the drive. The data is not stored in a linear manner on the disk; rather, it is stored in the best way for quickest retrieval.

The downside of traditional disk storage is that it takes time for the disk drives to rotate as they seek out the right sectors and then read off consecutive data pieces. To speed up the process of accessing data, vendors have introduced disk controllers that copy data from a disk into DRAM. These are called disk caches (similar to a cache used for main memory).

Solid-state drives

A solid-state drive (SSD) uses integrated circuit assemblies to store data persistently. SSDs have no moving (mechanical) components, and compared with electromechanical disks, they are typically more resistant to physical shock, run silently and have lower access time and less latency. However, while the price of SSDs has continued to decline over time, SSDs are still more expensive per unit of storage than hard disk drives (HDDs).

Most SSDs use NAND-based Flash memory, which retains data without power. For applications requiring fast access, but not necessarily data persistence after power loss, SSDs may be constructed from RAM. Such devices may employ separate power sources, such as batteries, to maintain data after power loss.

Hybrid drives or solid-state hybrid drives (SSHD) combine the features of SSDs and HDDs in the same unit, containing a large hard disk drive and a SSD cache to improve performance of frequently accessed data.

How disks and drives can be connected to hyperscale computingsystems

Disks and drives can be connected to hyperscale computing systems using several types of storage: network attached, storage-area network and tiered storage.

Network-attached storage (NAS)

Network-attached storage (NAS) is file-level data storage connected to a computer network that provides data access to multiple computers. Potential benefits of NAS over file servers include faster data access, easier administration and simpler configuration. NAS devices have gained popularity because they enable convenient file sharing among multiple computers.

NAS can take the form of a specialized computer built from the ground up specifically for storing and serving files—as opposed to a general-purpose computer used for this role. These systems contain one or more hard drives, often arranged into redundant arrays of inexpensive disks (RAIDs). NAS removes the responsibility of file serving from other servers on the network, providing access to files using network file sharing protocols.

Hard drives with "NAS" in their name are functionally similar to other drives. However, they may have different firmware, vibration tolerance or power dissipation to make them more suitable for use in RAIDs.

Storage-area network (SAN)

A storage-area network (SAN) provides access to consolidated, block-level data storage. SANs are used primarily to enhance storage devices accessible to servers so the devices seem locally attached to the operating system. A SAN typically has its own network of storage devices that other devices cannot access through the LAN.

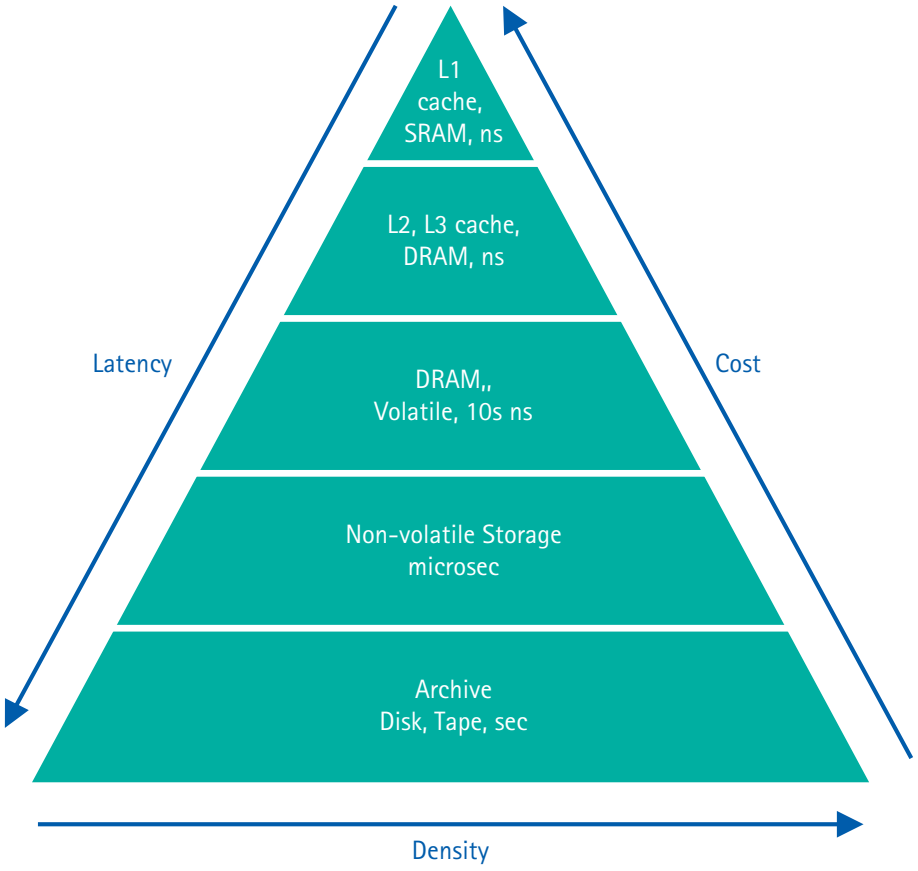
A SAN does not provide file-level access to data, only block-level operations. However, file systems built on top of SANs do provide file-level access. Such systems are known as SAN file systems or shared disk file systems.

Tiered storage

With tiered storage, data is stored in different memory formats in separate levels within a computer system. The levels differ in how quickly the CPU can access and conduct operations on the data (see Figure 7).

Level 1 caches stored in SRAM provide the fastest data access. However, storing data in this level is the most expensive option, in terms of cost per bit. In the future, newer technologies will be developed, and non-volatile memory will take up a larger share of the storage space in a computer system. Disk storage will move to archive.

Figure 7: Computer memory hierarchy.



Hierarchical storage management

Hierarchical storage management (HSM) is a data storage technique that automatically moves data between high-cost and low-cost storage media. HSM is typically performed by dedicated software. HSM systems store the bulk of the enterprise's data on slower devices, and then copy data to faster disk drives as needed. In effect, HSM turns the fast disk drives into caches for the slower mass-storage devices. The HSM system monitors the way data is used and makes best guesses as to which data can safely be moved to slower devices and which data should stay on the fast devices.

Recently, the development of serial advanced technology attachment (SATA) disks has created a market for three-stage HSM: files are migrated from high-performance fiber SAN devices to somewhat slower (but much cheaper) SATA disk arrays totaling several terabytes or more, and then eventually from the SATA disks to tape. But the newest development in HSM is the incorporation of hard disk drives and Flash, which is more than 30 times faster than disks.

Software-defined storage

Software-defined storage (SDS) technology separates storage hardware from software that manages the storage infrastructure. The software enabling an SDS environment provides policy management for feature options such as replication, snapshots and backup of data. By definition, SDS software is separate from the hardware it is managing. That hardware may or may not have abstraction, pooling or software embedded that can automate the migration of data from one disk to another, or one level to another. If SDS can be used on commodity servers with disks, it resembles software such as a file system. If it is layered over sophisticated, large storage arrays, it resembles software such as storage virtualization or storage resource management. These categories of products are positioned very differently in the marketplace.

Questions for your next meeting

What storage technologies are currently emphasized most in the computing systems your organization uses? What advantages and disadvantages do these technologies present?

How might advances in storage technologies, including tiered storage and hierarchical storage management, benefit your organization?

If you are considering establishing hardware infrastructure to support a private data center in your organization, what storage technologies would best support this effort?

Networking

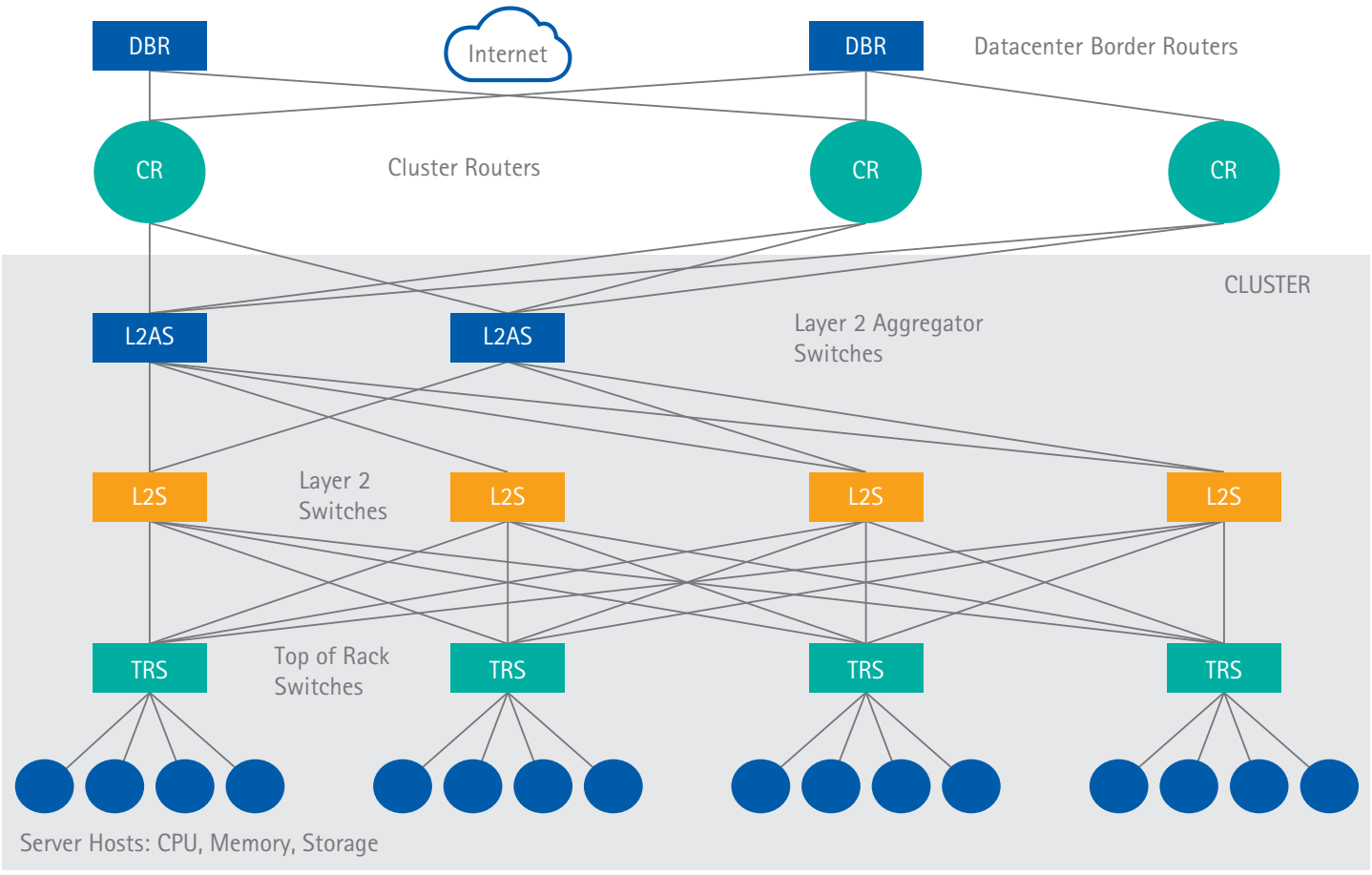
In this section, we look at the data-center networks on which hyperscale computing depends today—examining the architectures of these networks, along with additional network characteristics.

Data-center network architectures

A modern data center is home to tens of thousands of hosts, each consisting of one or more processors, memory, a network interface and local high-speed input/output (I/O). Compute resources are packaged into "racks" and allocated as clusters comprising thousands of hosts tightly connected with a high-bandwidth network (see Figure 8).

While the network plays a central role in overall system performance, it typically represents only 10-15 percent of the cluster cost. But take care not to confuse cost with value: the network is to a cluster computer as the central nervous system is to the human body.

Figure 8: Data-center network topology.



The thousands of hosts are orchestrated to help Internet workloads (such as millions of search requests carried out in a Google data center) divide incoming requests into parallel subtasks, and then weave together results from many subtasks across thousands of cores. In general, all parallel subtasks must complete in order for the request to complete. As a result, the maximum response time of any one subtask will dictate overall response time.

High-performing computer applications typically use the data center in a dedicated fashion to avoid contention from multiple applications and to reduce the resulting variation in application performance. By contrast, Web applications for scale-out hyperscale systems often rely on services sourced from multiple clusters, where each cluster may have several applications running simultaneously to increase overall system utilization. As a result, a data-center cluster may use virtualization (creation of virtual compute resources that are mapped onto physical computers to improve data center efficiency) for performance and fault isolation. Web applications are programmed with this sharing in mind.

To reduce the likelihood of congestion, the network can be overprovisioned with enough bandwidth even for traffic patterns that strain the network. But overprovisioning within large-scale networks is prohibitively expensive. Alternatively, implementing quality-of-service (QoS) policies to segregate traffic into distinct classes, and to provide performance isolation and high-level traffic engineering, can help ensure that application-level service level agreements (SLAs) are satisfied.

Network traffic

Understanding network traffic is essential to capacity planning and traffic engineering. Traffic in a data-center network is often measured and characterized according to flows—sequences of packets from a source to a destination host. In the realm of Internet protocols, a flow also includes a specific source and destination port number and transport type.

Network traffic is asymmetric, with client-to-server requests being abundant but generally small. However, server-to-client responses tend to be larger flows, though this, too, depends on the application. Moreover, Internet traffic becomes highly aggregated; as a result, the mean of traffic flows says very little because aggregated traffic exhibits a high degree of variability. Consequently, a network that is only 10 percent utilized can see lots of packet discards when running a Web search. Today's typical multitiered data-center network has a significant amount of oversubscription, where the hosts attached to the rack switch have significantly more provisioned bandwidth between one another than they do with hosts in other racks. This so-called rack affinity reduces network cost and improves utilization.

Between data centers, bandwidth is often very expensive over vast distances that have highly regulated traffic streams and patterns to ensure that expensive links are highly utilized. When congestion occurs, the most important traffic gets access to the links.

Network topology

A network's topology describes how switches and hosts are interconnected. Topology is central to a network's performance and cost, since it affects design tradeoffs among criteria such as performance, system packaging, path diversity, redundancy and resilience.

A fat-tree topology has an aggregate bandwidth that grows in proportion to the number of host ports in the system. In a scalable network, increasing the number of ports should linearly increase the delivered bandwidth. Scalability and reliability are inseparable, since growing to large system size requires a robust network.

Network addresses

In a network, messages are created by and delivered to endpoints. Thus, endpoints are distinguished from intermediate switching elements traversed en route. A host's address is how endpoints are identified in the network. The address can be thought of as the numerical equivalent of a host name.

An address is unique and must be represented in a canonical form that the routing function can use to determine where to send a packet. The switch inspects the packet header corresponding to the layer in which routing is performed. Address resolution protocols broadcast messages on the network to update local caches that are mapping layers. Switches assign IP addresses statically or disseminate host addresses using dynamic host configuration protocol (DHCP).

Gigabit Ethernet switches

In computer networking, the term gigabit Ethernet (GbE or 1 GigE) describes various technologies for transmitting Ethernet frames at a rate of a gigabit per second (Gbit/s) (1,000,000,000 bits per second), as defined by the IEEE 802.3-2008 standard. GbE cables and equipment closely resemble previous standards and have been common and economical since 2010. Higher bandwidth gigabit Ethernet standards have become available as the IEEE ratified new standards.

InfiniBand switches

InfiniBand® is a switched fabric computer network communications link used in high-performance computing and enterprise data centers. Its features include high throughput, low latency, quality of service and failover (by which traffic is routed to other switches and networks if a network fails). InfiniBand transmits data in packets that are taken together to form a message, such as a direct memory access Read from or Write to a remote node or a channel Send or Receive. InfiniBand switches are provided by vendors such as Mellanox.[10]

Like many other modern interconnects, InfiniBand offers point-to-point bidirectional serial links intended for the connection of processors with high-speed peripherals such as disks. InfiniBand also offers multicast operations, which support communication of one piece of data or information to multiple destinations. There are several possible signaling rates, and links can be bonded together for additional throughput. Larger systems with many more links are typically used for cluster and supercomputer interconnects and for inter-switch connections

The InfiniBand future roadmap calls for high data rate (HDR) to come in 2017. Next data rate (NDR) is due sometime later.

Silicon photonic interconnects

Silicon photonic devices can be made using existing semiconductor fabrication techniques. Because silicon is already used as the substrate for most integrated circuits, it is possible to create hybrid devices in which the optical and electronic components are integrated onto a single microchip. Consequently, many electronics manufacturers are actively researching silicon photonics for use in network interconnects.

The propagation of light through silicon devices is governed by a range of nonlinear optical phenomena. Nonlinearity enables light to interact with light, thus permitting applications such as wavelength conversion and all-optical signal routing.

Progress in computer technology depends increasingly on faster data transfer between and within microchips. Optical interconnects may provide a way forward, and silicon photonics may prove particularly useful, once integrated on the standard silicon chips. Complete transceivers have been commercialized in the form of active optical cables that have made advances in data-transfer speed. Graphene photodetectors, all-optical signal processing and silicon microphotonics may offer additional possibilities.

Software-defined networking

Software-defined networking (SDN) lets network administrators manage network services through abstraction of lower-level functionality. This is done by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). SDN requires some method for the control plane to communicate with the data plane.[11]

Elastic cloud architectures and dynamic allocation of resources, such as processors and storage to applications, are evolving. With these changes, and as usage of mobile computer operating systems and virtual machines grows, the need has arisen for an additional layer of SDN. Such a layer:

- Enables network operators to specify network services, without coupling these specifications with network interfaces. This enables applications and organizations to move between interfaces without changing identities or violating specifications.
- Simplifies network operations because global definitions for each identity do not have to be matched to every interface location.
- Reduces some of the complexity that has built up in network elements by decoupling identity and flow-specific control logic from basic topology-based forwarding of network packets to intermediate destinations, bridging across networks and routing.

SDN is a step in the evolution toward programmable and active networking. It enables network administrators to have programmable central control of network traffic via a controller without requiring physical access to the network's switches.

Questions for your next meeting

What networking technologies dominate your organization's current computing systems? What advantages and challenges do these technologies present for your organization?

What benefits and challenges might technologies that support faster networking—such as gigabit Ethernet and InfiniBand—present for your organization?

If you are considering establishing hardware architecture supporting a private data center for your organization, what networking technologies might make the best choices for your organization? Why?

Glossary

Cache: smaller, faster memory that stores copies of data from frequently used main memory locations.

Cluster: a collection of nodes, each of which consists of a processor, memory and storage.

Distributed processing: use of local-area networks (LANs) designed so a single program can run simultaneously at various sites.

Ethernet: a protocol for transferring information across computer networks.

Grain size: refers to the number of instructions executed in a processor before synchronizing, or communicating data, with another processor.

Hierarchical storage management: a data storage technique that automatically moves data between high-cost and low-cost storage media.

Instruction set architecture: specification of the types of instructions that a processor can execute and the format of the instructions.

Interconnection network: a network that connects processes in both shared- and distributed-memory parallel processing.

Multi-core processor: a single computing component with two or more independent CPUs ("cores"), which read and execute program instructions.

Network-attached storage: file-level data storage connected to a computer network that provides data access to multiple clients (computers).

Node: a set comprising a processor, memory and storage.

Non-volatile data: data that remains after power events or machine reboot cycles.

Non-volatile random access memory (NVRAM): RAM that retains its information when power is turned off.

Parallel processing: the simultaneous use of more than one CPU (processor) to execute a program.

Power usage efficiency: the ratio of total amount of energy used by a computer data center (including energy for lighting and cooling) to the energy delivered to the computing equipment.

Register: a fast local storage area on a computer's CPU.

Solid-state drive: a data storage device using integrated circuit assemblies as memory to store data persistently. Also known as a solid-state disk or electronic disk, though it contains no actual disk.

Storage-area network: a dedicated network that provides access to consolidated, block-level data storage.

References

[1] Nvidia Tesla K40 GPU, www.nvidia.com.

[2] www.opencompute.org.

[3] http://www.opencompute.org/about/open-compute-project-solution-providers.

[4] www.sgi.com.

[5] www.cray.com.

[6] www.top500.org.

[7] http://www.cresta-project.eu/the-exascale-challenge.html.

[8] www.openmp.org

[9] http://www.nvidia.com/object/cuda_home_new.html#sthash.jwZKUP2V.dpuf.

[10] http://www.mellanox.com/page/switch_systems_overview.

[11] https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

Contact us

Prith Banerjee
prithviraj.banerjee@accenture.com

Steven C. Tiell
steven.c.tiell@accenture.com

About Accenture

Accenture is a global management consulting, technology services and outsourcing company, with more than 323,000 people serving clients in more than 120 countries. Combining unparalleled experience, comprehensive capabilities across all industries and business functions, and extensive research on the world's most successful companies, Accenture collaborates with clients to help them become high-performance businesses and governments. The company generated net revenues of US$30.0 billion for the fiscal year ended Aug. 31, 2014. Its home page is www.accenture.com.

About Accenture Technology Labs

Accenture Technology Labs, the dedicated technology research and development (R&D) organization within Accenture, has been turning technology innovation into business results for more than 20 years. Our R&D team explores new and emerging technologies to create a vision of how technology will shape the future and invent the next wave of cutting-edge business solutions. Working closely with Accenture's global network of specialists, Accenture Technology Labs help clients innovate to achieve high performance. The Labs are located in Silicon Valley, California; Sophia Antipolis, France; Arlington, Virginia; Beijing, China and Bangalore, India. For more information, please visit www.accenture.com/technologylabs.

15-0438U/9-9443