Enabling Technologies for Distributed Computing

Dr. Sanjay P. Ahuja, Ph.D.
Fidelity National Financial Distinguished Professor of CIS
School of Computing, UNF
Technologies for Network-Based Systems

- **Multi-core CPUs and Multithreading Technologies**

  CPU's today assume a multi-core architecture with dual, quad, six, or more processing cores. The clock rate increased from 10 MHz for Intel 286 to 4 GHz for Pentium 4 in 30 years. However, the clock rate reached its limit on CMOS chips due to power limitations. Clock speeds cannot continue to increase due to excessive heat generation and current leakage.

  ![Historical CPU Clock Rates](image)

  Multi-core CPUs can handle multiple instruction threads.
Technologies for Network-Based Systems

- Multi-core CPUs and Multithreading Technologies

LI cache is private to each core, L2 cache is shared and L3 cache or DRAM is off the chip. Examples of multi-core CPUs include Intel i7, Xeon, AMD Opteron. Each core can also be multithreaded. E.g. the Niagara II has 8 cores with each core handling 8 threads for a total of 64 threads maximum.
Technologies for Network-Based Systems

- Memory, Storage and WAN
  - DRAM chip capacity increased from 16 KB in 1976 to 64 GB in 2011 for a 4x increase in capacity every 3 years. Memory access time did not improve as much.
  - For hard drives, capacity increased from 260 MB in 1981 to 3 TB for the Seagate Barracuda XT hard drive in 2011 for an approximate 10x increase in capacity every 8 years.
  - The "memory wall" is the growing disparity of speed between CPU and memory outside the CPU chip. An important reason for this disparity is the limited communication bandwidth beyond chip boundaries. From 1986 to 2000, CPU speed improved at an annual rate of 55% while memory speed only improved at 10%.
  - Faster processor speed and larger memory capacity result in a wider performance gap between processors and memory. The memory wall may become an even worse problem limiting CPU performance.
Technologies for Network-Based Systems

- System-Area Interconnects

- A LAN is typically used to connect clients to big servers. A Storage Area Network (SAN) connects servers to network storage such as disk arrays. Network attached storage (NAS) connects servers directly to disk arrays. All 3 types of networks often appear in a large cluster built with commercial network components.
Technologies for Network-Based Systems

- **System-Area Interconnects - NAS**
  - A NAS is fundamentally a bunch of disks, usually arranged in a disk array. Users and servers attach to the NAS primarily using TCP/IP over Ethernet, and the NAS has its own IP address. The primary job of a NAS is to serve files, so most NAS systems offer support for Windows networking, HTTP, plus file systems and protocols such as NFS.
Technologies for Network-Based Systems

- **System-Area Interconnects - SAN**
  - SANs allow multiple servers to share a RAID, making it appear to the server as if it were local or directly attached storage, and it cannot be accessed by individual users. A dedicated networking standard, Fiber Channel, allows blocks to be moved between servers and storage at high speed. It uses dedicated switches and a fiber-based cabling system which separates it from the day-to-day traffic. It uses the SCSI protocol for communication.
Technologies for Network-Based Systems

- Virtual Machines and Virtualization Middleware
  - To build clouds we need to aggregate large amounts of computing, storage, and networking resources in a virtualized manner. Specifically, clouds rely on the dynamic virtualization of CPU, memory, and I/O.
  - Virtual Machines (VMs)
    The VM is built with virtual resources managed by a guest OS to run a specific application.
    Between the VMs and the host platform, a middleware layer (called the Virtual Memory Monitor (VMM) or a hypervisor) is deployed.
  - Type 1 (bare metal) hypervisor
    The bare metal hypervisor runs on the bare hardware and handles all the Hardware (CPU, memory, and I/O) directly. This runs in the privileged mode. The guest OS could any OS such as Linux, Windows etc.
    They provide an almost native performance to the guest OSs (VMs), generally losing only 3–4% of the Central Processing Unit’s cycles to the running of the hypervisor.
Technologies for Network-Based Systems

- Virtual Machines and Virtualization Middleware

- Type 1 (hosted) hypervisor (continued)
- Bare-metal is great for consolidating a company’s collection of servers onto a single piece of hardware.
- Some examples of the leading bare-metal hypervisors are VMware’s ESX(i) (proprietary), Citrix Xen Server (FOS (Free & Open Source)), and KVM (kernel loaded VM) (FOS). ESX and XenServer are installs that reside directly on the hardware. KVM sits within a Linux kernel.
Technologies for Network-Based Systems

- Virtual Machines and Virtualization Middleware

- **Type 2 (or hosted) hypervisor**: Here the hypervisor runs on top of a host OS in user or non-privileged mode. The host OS need not be modified. For example, you could install Windows or Linux, and then install the hypervisor on top but the performance may not be as good as with bare-metal.

- Hosted is often used by IT workers who need the flexibility to install, run and try out different OSs on their own computers without disrupting their current computing environment.

- Some examples of the leading hosted hypervisors are VirtualBox (FOS) and Qemu (FOS)

- Many VMs can be run on a hypervisor. The resource most in demand is system memory, and because RAM is cheap, this makes the proposition of virtualization an attractive one.

- A VM can be suspended and stored in secondary storage, resumed, or migrated from one hardware platform to another.
Technologies for Network-Based Systems

- Full Virtualization vs. Para-Virtualization
  - **Full Virtualization** allows the guest OS to run on the hypervisor without any modification and without it knowing that it is hosted.
  - Paravirtualization requires that the kernel of the guest OS is modified and compiled with hooks (an API) for the hypervisor (guest OSs must know about the hypervisor). The guest OS can then communicates and cooperates with the hypervisor with a potential to improving performance, though this might be marginal (load that generates system calls benefits the most).
- Windows guests can only run on Full Virtualization, as their source is proprietary.
- Operating systems that support paravirtualization interfaces need custom kernel adjustments. If compiled manually (where possible), guest operating systems with hypervisor support require more maintenance and configuration. This additional costs and complexity, combined with the marginal performance gains, means paravirtualization remains a niche product in the server virtualization market.