Cluster Computing

Introduction

Cluster Computing at a Glance

Introduction

- Need more computing power
  - Improve the operating speed of processors & other components
    - constrained by the speed of light, thermodynamic laws, & the high financial costs for processor fabrication
  - Connect multiple processors together & coordinate their computational efforts
    - parallel computers
    - allow the sharing of a computational task among multiple processors

2

How to Run Applications Faster ?

- There are 3 ways to improve performance:
  - Work Harder
  - Work Smarter
  - Get Help
- Computer Analogy
  - Using faster hardware
  - Optimized algorithms and techniques used to solve computational tasks
  - Multiple computers to solve a particular task

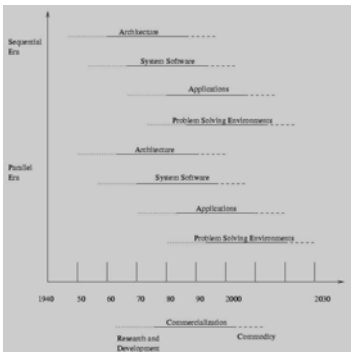
3

Era of Computing

- Rapid technical advances
  - the recent advances in VLSI technology
  - software technology
    - OS, PL, development methodologies, & tools
  - grand challenge applications have become the main driving force
- Parallel computing
  - one of the best ways to overcome the speed bottleneck of a single processor
  - good price/performance ratio of a small cluster-based parallel computer

4

Two Eras of Computing



5

Scalable Parallel Computer Architectures

- Taxonomy - based on how processors, memory & interconnect are laid out
  - Massively Parallel Processors (MPP)
  - Symmetric Multiprocessors (SMP)
  - Cache-Coherent Non-Uniform Memory Access (CC-NUMA)
  - Distributed Systems
  - Clusters

6

Scalable Parallel Computer Architectures

- MPP
  - A large parallel processing system with a shared-nothing architecture
  - Consist of several hundred nodes with a high-speed interconnection network/switch
  - Each node
    - consists of a main memory & one or more processors
    - runs a separate copy of the OS
- SMP
  - 2-64 processors today
  - Shared-everything architecture
  - All processors share all the global resources available
  - Single copy of the OS runs on these systems

7

Scalable Parallel Computer Architectures

- CC-NUMA
  - a scalable multiprocessor system having a cache-coherent nonuniform memory access architecture
  - every processor has a global view of all of the memory
- Distributed systems
  - considered conventional networks of independent computers
  - the individual machines could be combinations of MPPs, SMPs, clusters, & individual computers
  - have multiple system images as each node runs its own OS
- Clusters
  - a collection of workstations or PCs that are interconnected by a high-speed network
  - work as an integrated collection of resources
  - have a single system image spanning all its nodes

8

Key Characteristics of Scalable Parallel Computers

Characteristic	MPP	SMP CC-NUMA	Cluster	Distributed
System Number of Nodes	O(100) - O(1000)	O(10) - O(100)	O(100) or less	O(10) - O(1000)
Node Complexity	Fine or medium grain	Medium or coarse grain	Medium grain	Wide range
Inter-node Communication	Message passing or shared variables for DSM	Centralized and distributed shared memory	Message Passing	Shared files RPC message passing IPC protocol
Job Scheduling	Single run queue on host	Single run queue mostly	Multiple queues but coordinated	Independent multiple Queues
SSI Support	Partially	Always in SMP and some NUMA	Desired	No
Node OS Copies and Type	N Micro-kernels, monolithic, or layered OSs	One monolithic for SMP and multiple for NUMA	N OS platforms (Homogeneous or micro-kernel)	N OS platforms (Heterogeneous)
Address Space	Multiple (Single for DSM)	Single	Multiple or single	Multiple
Inter-node Security	Unnecessary	Unnecessary	Required if exposed	Required
Ownership	One organization	One organization	One or More organizations	Many organizations

9

Towards Low Cost Parallel Computing

- Parallel processing
  - linking together 2 or more computers to jointly solve some computational problem
  - since the early 1990s, an increasing trend to move away from expensive and specialized proprietary parallel supercomputers towards networks of workstations
  - the rapid improvement in the availability of commodity high performance components for workstations and networks
    - Low-cost commodity supercomputing
  - from specialized traditional supercomputing platforms to cheaper, general purpose systems consisting of loosely coupled components built up from single or multiprocessor PCs or workstations
  - need to standardization of many of the tools and utilities used by parallel applications (eg. MPI, HPF)

10

Motivations of using NoW over Specialized Parallel Computers

- Individual workstations are becoming increasing powerful
- Communication bandwidth between workstations is increasing and latency is decreasing
- Workstation clusters are easier to integrate into existing networks
- Typical low user utilization of personal workstations
- Development tools for workstations are more mature
- Workstation clusters are a cheap and readily available
- Clusters can be easily grown

11

Trend

- Workstations with UNIX for science & industry vs. PC-based machines for administrative work & work processing
- A rapid convergence in processor performance and kernel-level functionality of UNIX workstations and PC-based machines

12

Windows of Opportunities

- Parallel Processing
  - Use multiple processors to build MPP/DSM-like systems for parallel computing
- Network RAM
  - Use memory associated with each workstation as aggregate DRAM (Dynamic RAM) cache
- Software RAID
  - Redundant array of inexpensive disks
  - Use the arrays of workstation disks to provide cheap, highly available, & scalable file storage
  - Possible to provide parallel I/O support to applications
  - Use arrays of workstation disks to provide cheap, highly available, and scalable file storage
- Multipath Communication
  - Use multiple networks for parallel data transfer between nodes

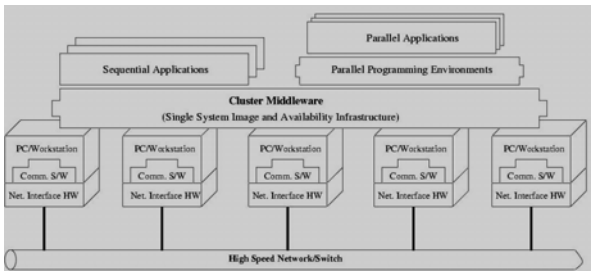
13

Cluster Computer and its Architecture

- A cluster is a type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers cooperatively working together as a single, integrated computing resource
- A node
  - a single or multiprocessor system with memory, I/O facilities, & OS
  - generally 2 or more computers (nodes) connected together
  - in a single cabinet, or physically separated & connected via a LAN
  - appear as a single system to users and applications
  - provide a cost-effective way to gain features and benefits

14

Cluster Computer Architecture



15

Prominent Components of Cluster Computers

- Multiple High Performance Computers
  - Cluster of PCs (CoPs), Pile of PCs (PoPs)
  - Cluster of Workstations (CoWs), Network of Workstations (NoWs)
  - Cluster of SMPs (CLUMPS)
  - Distributed HPC Systems leading to Metacomputing

16

Prominent Components of Cluster Computers

- State of the art Operating Systems
  - Linux (Beowulf)
  - Microsoft (Illinois HPVM)
  - SUN Solaris (Berkeley NOW)
  - IBM AIX (IBM SP2)
  - HP UX (Illinois - PANDA)
  - Mach (Microkernel based OS, CMU)
  - Cluster Operating Systems (Solaris MC, MOSIX - academic project)
  - OS gluing layers (Berkeley GLUnix)

17

Prominent Components of Cluster Computers

- High Performance Networks/Switches
  - Fast Ethernet (100Mbps),
  - Gigabit Ethernet (1Gbps)
  - SCI (Scalable Coherent Interface) e.g. by Dolphin: MPI with 12µsec latency
  - ATM
  - Myrinet (1.2Gbps)
  - Digital Memory Channel
  - FDDI

18

Prominent Components of Cluster Computers

- Network Interface Card
  - Myrinet has NIC
  - User-level access support

19

Prominent Components of Cluster Computers

- Fast Communication Protocols and Services,

  e.g.
  - Active Messages (Berkeley)
  - Fast Messages (Illinois)
  - U-net (Cornell)
  - XTP (Virginia)

20

Prominent Components of Cluster Computers

- Cluster Middleware
  - Single System Image (SSI)
  - System Availability (SA) Infrastructure
- Hardware
  - DEC Memory Channel, DSM (Alewife, DASH), SMP Techniques
- Operating System Kernel/Gluing Layers
  - Solaris MC, SCO Unixware, GLUnix
- Applications and Subsystems
  - Applications (system management and electronic forms)
  - Runtime systems (software DSM, PFS etc.)
  - Resource management and scheduling software (RMS)
    - CODINE, LSF, PBS, NQS, etc.

21

Prominent Components of Cluster Computers

- Parallel Programming Environments and Tools
  - Threads (PCs, SMPs, NOW..)
    - POSIX Threads
    - Java Threads
  - MPI
    - Linux, NT, on many Supercomputers
  - PVM
  - Software DSMs (ShMem)
  - Compilers
    - C/C++/Java
    - Parallel programming with C++ (MIT Press book)
  - RAD (rapid application development tools)
    - GUI based tools for PP modeling
  - Debuggers
  - Performance Analysis Tools
  - Visualization Tools

22

Prominent Components of Cluster Computers

- Applications
  - Sequential
  - Parallel / Distributed (Cluster-aware app.)
    - Grand Challenging applications
      - Weather Forecasting
      - Quantum Chemistry
      - Molecular Biology Modeling
      - Engineering Analysis (CAD/CAM)
      - . . .
    - Parallel DBMSs, web servers, data-mining

23

Key Operational Benefits of Clustering

- High Performance
- Expandability and Scalability
- High Throughput
- High Availability

24

Clusters Classification

- Application Target
  - High Performance (HP) Clusters
    - Grand Challenging Applications
  - High Availability (HA) Clusters
    - Mission Critical applications

25

Clusters Classification

- Node Ownership
  - Dedicated Clusters
  - Non-dedicated clusters
    by stealing idle CPU cycles as most workstation CPU cycles are unused, even during peak hours
    - Adaptive parallel computing
      parallel computing on a dynamically changing set of nondedicated workstations
    - Communal multiprocessing

26

Clusters Classification

- Node Hardware
  - Clusters of PCs (CoPs), Piles of PCs (PoPs)
  - Clusters of Workstations (CoWs)
  - Clusters of SMPs (CLUMPs)

27

Clusters Classification

- Node Operating System
  - Linux Clusters (e.g., Beowulf)
  - Solaris Clusters (e.g., Berkeley NOW)
  - NT Clusters (e.g., HPVM)
  - AIX Clusters (e.g., IBM SP2)
  - SCO/Compaq Clusters (Unixware)
  - Digital VMS Clusters
  - HP-UX clusters
  - Microsoft Wolfpack clusters

28

Clusters Classification

- Node Configuration
  - Homogeneous Clusters
    - All nodes will have similar architectures and run the same OSs
  - Heterogeneous Clusters
    - All nodes will have different architectures and run different OSs

29

Clusters Classification

- Levels of Clustering
  - Group Clusters (#nodes: 2-99)
    - Nodes are connected by SAN like Myrinet
  - Departmental Clusters (#nodes: 10s to 100s)
  - Organizational Clusters (#nodes: many 100s)
  - National Metacomputers (WAN/Internet-based)
  - International Metacomputers (Internet-based, #nodes: 1000s to many millions)
    - Metacomputing
    - Web-based Computing
    - Agent Based Computing

"Java plays a major in web and agent based computing"

30

Commodity Components for Clusters

- Processors
  - Intel x86 Processors
    - Pentium, Pentium Xeon, AMD x86, Cyrix x86, etc.
  - Digital Alpha
    - Alpha 21364 processor integrates processing, memory controller, network interface into a single chip
  - IBM PowerPC
  - Sun SPARC
  - SGI MIPS
  - HP PA
  - Berkeley Intelligent RAM (IRAM) integrates processor and DRAM onto a single chip

31

Commodity Components for Clusters

- Memory and Cache
  - Access to DRAM is extremely slow compared to the speed of the processor
    - the very fast memory used for Cache is expensive & cache control circuitry becomes more complex as the size of the cache grows
  - Within Pentium-based machines, uncommon to have a 64-bit wide memory bus as well as a chip set that support 2Mbytes of external cache

32

Commodity Components for Clusters

- Disk and I/O
  - Overall improvement in disk access time has been less than 10% per year
  - Amdahl's law
    - Speed-up obtained by from faster processors is limited by the slowest system component
  - Parallel I/O
    - Carry out I/O operations in parallel, supported by parallel file system based on hardware or software RAID

33

Commodity Components for Clusters

- System Bus
  - PCI bus
    - 133Mbytes/s transfer rate
    - Adopted both in Pentium-based PC and non-Intel platform (e.g., Digital Alpha Server)

34

Commodity Components for Clusters

- Cluster Interconnects
  - Communicate over high-speed networks using a standard networking protocol such as TCP/IP or a low-level protocol such as AM (Active Messaging)
  - Fast Ethernet and Gigabit Ethernet
    - Fast Ethernet – 100 Mbps
    - Gigabit Ethernet
      - preserve Ethernet's simplicity
      - deliver a very high bandwidth to aggregate multiple Fast Ethernet segments

35

Commodity Components for Clusters

- Cluster Interconnects
  - Asynchronous Transfer Mode (ATM)
    - Switched virtual-circuit technology
    - Cell (small fixed-size data packet)
    - use optical fiber - expensive upgrade
    - telephone style cables (CAT-3) & better quality cable (CAT-5)

36

Commodity Components for Clusters

- Cluster Interconnects
  - Scalable Coherent Interfaces (SCI)
    - IEEE 1596-1992 standard aimed at providing a low-latency distributed shared memory across a cluster
    - Point-to-point architecture
      - reduce the delay interprocessor communication
      - eliminate the need for runtime layers of software protocol-paradigm translation
      - less than 12 μsec zero message-length latency on Sun SPARC
    - Support distributed multiprocessing with high bandwidth and low latency
    - SCI cards for SPARC's SBus and PCI-based SCI cards from Dolphin
    - Scalability constrained by the current generation of switches & relatively expensive components

37

Commodity Components for Clusters

- Cluster Interconnects
  - Myrinet
    - 1.28 Gbps full duplex interconnection network
    - Use low latency cut-through routing switches, which is able to offer fault tolerance by automatic mapping of the network configuration
    - Support both Linux & Windows
    - Advantages
      - Very low latency (5μs, one-way point-to-point)
      - Very high throughput
      - Programmable on-board processor for greater flexibility
    - Disadvantages
      - Expensive: $1500 per host
      - Complicated scaling: switches with more than 16 ports are unavailable

38

Commodity Components for Clusters

- Operating Systems
  - 2 fundamental services for users
    - make the computer hardware easier to use
      - create a virtual machine that differs markedly from the real machine
    - share hardware resources among users
      - processor - multitasking
  - The new concept in OS services
    - support multiple threads of control in a process itself
      - parallelism within a process
      - multithreading
      - POSIX thread interface is a standard programming environment
  - Trend
    - Modularity – MS Windows, IBM OS/2
    - Microkernel – provide only essential OS services
      - high level abstraction of OS portability

39

Commodity Components for Clusters

- Operating Systems
  - Linux
    - UNIX-like OS
    - Runs on cheap x86 platform, yet offers the power and flexibility of UNIX
    - Readily available on the Internet and can be downloaded without cost
    - Easy to fix bugs and improve system performance
    - Users can develop or fine-tune hardware drivers which can easily be made available to other users
    - Features such as preemptive multitasking, demand-page virtual memory, multiuser, multiprocessor support

40

Commodity Components for Clusters

- Operating Systems
  - Solaris
    - UNIX-based multithreading and multiuser OS
    - support Intel x86 & SPARC-based platforms
    - Real-time scheduling feature critical for multimedia applications
    - Support two kinds of threads
      - Light Weight Processes (LWPs)
      - User level thread
    - Support both BSD and several non-BSD file system
      - CacheFS
      - AutoClient
      - TmpFS: uses main memory to contain a file system
      - *Proc* file system
      - *Volume* file system
    - Support distributed computing
    - Able to store & retrieve distributed information
    - OpenWindows allows application to be run on remote systems

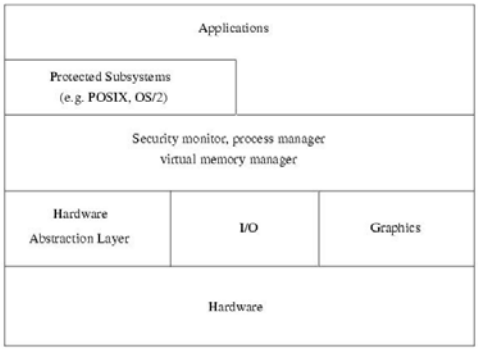
41

Commodity Components for Clusters

- Operating Systems
  - Microsoft Windows
    - Preemptive, multitasking, multiuser, 32-bits OS
    - Object-based security model and special file system (NTFS) that allows permissions to be set on a file and directory basis
    - Support multiple CPUs and provide multitasking using symmetrical multiprocessing
    - Support different CPUs and multiprocessor machines with threads
    - Have the network protocols & services integrated with the base OS
      - several built-in networking protocols (IPX/SPX., TCP/IP, NetBEUI), & APIs (NetBIOS, DCE RPC, Window Sockets (Winsock))

42

Windows Architecture



43

Network Services/ Communication SW

- Communication infrastructure support protocol for
  - Bulk-data transport
  - Streaming data
  - Group communications
- Communication service provide cluster with important QoS parameters
  - Latency
  - Bandwidth
  - Reliability
  - Fault-tolerance
  - Jitter control
- Network service are designed as hierarchical stack of protocols with relatively low-level communication API, provide means to implement wide range of communication methodologies
  - RPC
  - DSM
  - Stream-based and message passing interface (e.g., MPI, PVM)

44

Cluster Middleware & SSI

- SSI
  - Supported by a middleware layer that resides between the OS and user-level environment
  - Middleware consists of essentially 2 sublayers of SW infrastructure
    - SSI infrastructure
      - Glue together OSs on all nodes to offer unified access to system resources
    - System availability infrastructure
      - Enable cluster services such as checkpointing, automatic failover, recovery from failure, & fault-tolerant support among all nodes of the cluster

45

What is Single System Image (SSI) ?

- A single system image is the illusion, created by software or hardware, that presents a collection of resources as one, more powerful resource.
- SSI makes the cluster appear like a single machine to the user, to applications, and to the network.
- A cluster without a SSI is not a cluster

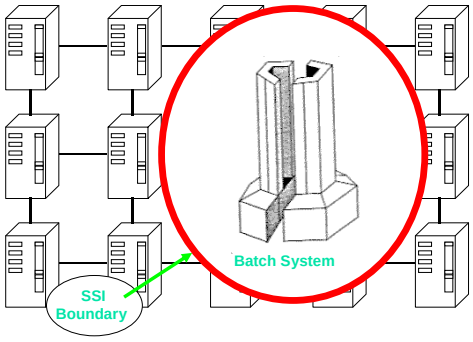
46

Single System Image Boundaries

- Every SSI has a boundary
- SSI support can exist at different levels within a system, one able to be build on another

47

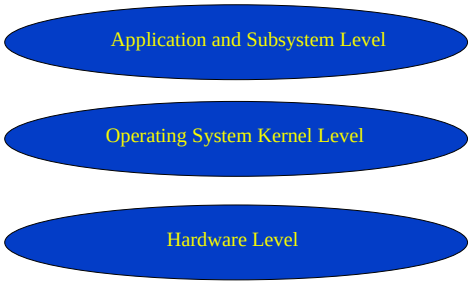
SSI Boundaries -- an applications SSI boundary



(c) In search of clusters

48

SSI Levels/Layers



49

SSI at Hardware Layer

Level	Examples	Boundary	Importance
Application and Subsystem Level			
Operating System Kernel Level			
memory	SCI, DASH	memory space	better communication and synchronization
memory and I/O	SCI, SMP techniques	memory and I/O device space	lower overhead cluster I/O

50 (c) In search of clusters

SSI at Operating System Kernel (Underware) or Gluing Layer

Level	Examples	Boundary	Importance
Kernel/ OS Layer	Solaris MC, Unixware MOSIX, Sprite,Amoeba / GLUnix	each name space: files, processes, pipes, devices, etc.	kernel support for applications, adm subsystems
kernel interfaces	UNIX (Sun) vnode, Locus (IBM) vproc	type of kernel objects: files, processes, etc.	modularizes SSI code within kernel
virtual memory	none supporting operating system kernel	each distributed virtual memory space	may simplify implementation of kernel objects
microkernel	Mach, PARAS, Chorus, OSF/1AD, Amoeba	each service outside the microkernel	implicit SSI for all system services

51 (c) In search of clusters

SSI at Application and Subsystem Layer (Middleware)

Level	Examples	Boundary	Importance
application	cluster batch system, system management	an application	what a user wants
subsystem	distributed DB, OSF DME, Lotus Notes, MPI, PVM	a subsystem	SSI for all applications of the subsystem
file system	Sun NFS, OSF, DFS, NetWare, and so on	shared portion of the file system	implicitly supports many applications and subsystems
toolkit	OSF DCE, Sun ONC+, Apollo Domain	explicit toolkit facilities: user, service name,time	best level of support for heter-ogeneous system

52 (c) In search of clusters

Single System Image Benefits

- Provide a simple, straightforward view of all system resources and activities, from any node of the cluster
- Free the end user from having to know where an application will run
- Free the operator from having to know where a resource is located
- Let the user work with familiar interface and commands and allows the administrators to manage the entire clusters as a single entity
- Reduce the risk of operator errors, with the result that end users see improved reliability and higher availability of the system

53

Single System Image Benefits (Cont'd)

- Allowing centralize/decentralize system management and control to avoid the need of skilled administrators from system administration
- Present multiple, cooperating components of an application to the administrator as a single application
- Greatly simplify system management
- Provide location-independent message communication
- Help track the locations of all resource so that there is no longer any need for system operators to be concerned with their physical location
- Provide transparent process migration and load balancing across nodes.
- Improved system response time and performance

54

Middleware Design Goals

- Complete Transparency in Resource Management
  - Allow user to use a cluster easily without the knowledge of the underlying system architecture
  - The user is provided with the view of a globalized file system, processes, and network
- Scalable Performance
  - Can easily be expanded, their performance should scale as well
  - To extract the max performance, the SSI service must support load balancing & parallelism by distributing workload evenly among nodes
- Enhanced Availability
  - Middleware service must be highly available at all times
  - At any time, a point of failure should be recoverable without affecting a user's application
    - Employ checkpointing & fault tolerant technologies
  - Handle consistency of data when replicated

55

SSI Support Services

- Single Entry Point
  - telnet cluster.myinstitute.edu
  - telnet node1.cluster. myinstitute.edu
- Single File Hierarchy: xFS, AFS, Solaris MC Proxy
- Single Management and Control Point: Management from single GUI
- Single Virtual Networking
- Single Memory Space - Network RAM / DSM
- Single Job Management: GLUnix, Codine, LSF
- Single User Interface: Like workstation/PC windowing environment (CDE in Solaris/NT), may it can use Web technology

56

Availability Support Functions

- Single I/O Space (SIOS):
  - any node can access any peripheral or disk devices without the knowledge of physical location.
- Single Process Space (SPS)
  - Any process on any node create process with cluster wide process wide and they communicate through signal, pipes, etc, as if they are one a single node.
- Checkpointing and Process Migration.
  - Saves the process state and intermediate results in memory to disk to support rollback recovery when node fails
  - PM for dynamic load balancing among the cluster nodes

57

Resource Management and Scheduling (RMS)

- RMS is the act of distributing applications among computers to maximize their throughput
- Enable the effective and efficient utilization of the resources available
- Software components
  - Resource manager
    - Locating and allocating computational resource, authentication, process creation and migration
  - Resource scheduler
    - Queueing applications, resource location and assignment
- Reasons using RMS
  - Provide an increased, and reliable, throughput of user applications on the systems
  - Load balancing
  - Utilizing spare CPU cycles
  - Providing fault tolerant systems
  - Manage access to powerful system, etc
- Basic architecture of RMS: client-server system

58

Services provided by RMS

- Process Migration
  - Computational resource has become too heavily loaded
  - Fault tolerant concern
- Checkpointing
- Scavenging Idle Cycles
  - 70% to 90% of the time most workstations are idle
- Fault Tolerance
- Minimization of Impact on Users
- Load Balancing
- Multiple Application Queues

59

Some Popular Resource Management Systems

Project	Commercial Systems - URL
LSF	http://www.platform.com/
CODINE	http://www.genias.de/products/codine/tech_desc.html
Easy-LL	http://www.tc.cornell.edu/UserDoc/SP/LL12/Easy/
NQE	http://www.cray.com/products/software/nqe/
	Public Domain System - URL
CONDOR	http://www.cs.wisc.edu/condor/
GNQS	http://www.gnqs.org/
DQS	http://www.scri.fsu.edu/~pasko/dqs.html
PRM	http://gost.isi.edu/gost-group/products/prm/
PBS	http://pbs.mrj.com/

60

Programming Environments and Tools

- Threads (PCs, SMPs, NOW..)
  - In multiprocessor systems
    - Used to simultaneously utilize all the available processors
  - In uniprocessor systems
    - Used to utilize the system resources effectively
  - Multithreaded applications offer quicker response to user input and run faster
  - Potentially portable, as there exists an IEEE standard for POSIX threads interface (pthreads)
  - Extensively used in developing both application and system software

61

Programming Environments and Tools

- Message Passing Systems (MPI and PVM)
  - Allow efficient parallel programs to be written for distributed memory systems
  - 2 most popular high-level message-passing systems – PVM & MPI
  - PVM
    - both an environment & a message-passing library
  - MPI
    - a message passing specification, designed to be standard for distributed memory parallel computing using explicit message passing
    - attempt to establish a practical, portable, efficient, & flexible standard for message passing
    - generally, application developers prefer MPI, as it is fast becoming the de facto standard for message passing

62

Programming Environments and Tools

- Distributed Shared Memory (DSM) Systems
  - Message-passing
    - the most efficient, widely used, programming paradigm on distributed memory system
    - complex & difficult to program
  - Shared memory systems
    - offer a simple and general programming model
    - but suffer from scalability
  - DSM on distributed memory system
    - alternative cost-effective solution
    - Software DSM
      - Usually built as a separate layer on top of the comm interface
      - Take full advantage of the application characteristics: virtual pages, objects, & language types are units of sharing
      - ThreadMarks, Linda
    - Hardware DSM
      - Better performance, no burden on user & SW layers, fine granularity of sharing, extensions of the cache coherence scheme, & increased HW complexity
      - DASH, Merlin

63

Programming Environments and Tools

- Parallel Debuggers and Profilers
  - Debuggers
    - Very limited
    - HPDF (High Performance Debugging Forum) as Parallel Tools Consortium project in 1996
      - Developed a HPD version specification, which defines the functionality, semantics, and syntax for a commercial-line parallel debugger
  - TotalView
    - A commercial product from Dolphin Interconnect Solutions
    - The only widely available GUI-based parallel debugger that supports multiple HPC platforms
    - Only used in homogeneous environments, where each process of the parallel application being debugged must be running under the same version of the OS

64

Functionality of Parallel Debugger

- Managing multiple processes and multiple threads within a process
- Displaying each process in its own window
- Displaying source code, stack trace, and stack frame for one or more processes
- Diving into objects, subroutines, and functions
- Setting both source-level and machine-level breakpoints
- Sharing breakpoints between groups of processes
- Defining watch and evaluation points
- Displaying arrays and its slices
- Manipulating code variable and constants

65

Programming Environments and Tools

- Performance Analysis Tools
  - Help a programmer to understand the performance characteristics of an application
  - Analyze & locate parts of an application that exhibit poor performance and create program bottlenecks
  - Major components
    - A means of inserting instrumentation calls to the performance monitoring routines into the user's applications
    - A run-time performance library that consists of a set of monitoring routines
    - A set of tools for processing and displaying the performance data
  - Issue with performance monitoring tools
    - Intrusiveness of the tracing calls and their impact on the application performance
    - Instrumentation affects the performance characteristics of the parallel application and thus provides a false view of its performance behavior

66

Performance Analysis and Visualization Tools

Tool	Supports	URL
AIMS	Instrumentation, monitoring library, analysis	http://science.nas.nasa.gov/Software/AIMS
MPE	Logging library and snapshot performance visualization	http://www.mcs.anl.gov/mpi/mpich
Pablo	Monitoring library and analysis	http://www-pablo.cs.uiuc.edu/Projects/Pablo/
Paradyn	Dynamic instrumentation running analysis	http://www.cs.wisc.edu/paradyn
SvPablo	Integrated instrumentor, monitoring library and analysis	http://www-pablo.cs.uiuc.edu/Projects/Pablo/
Vampir	Monitoring library performance visualization	http://www.pallas.de/pages/vampir.htm
Dimenmas	Performance prediction for message passing programs	http://www.pallas.com/pages/dimemas.htm
Paraver	Program visualization and analysis	http://www.cepba.upc.es/paraver

67

Programming Environments and Tools

- Cluster Administration Tools
  - Berkeley NOW
    - Gather & store data in a relational DB
    - Use Java applet to allow users to monitor a system
  - SMILE (Scalable Multicomputer Implementation using Low-cost Equipment)
    - Called K-CAP
    - Consist of compute nodes, a management node, & a client that can control and monitor the cluster
    - K-CAP uses a Java applet to connect to the management node through a predefined URL address in the cluster
  - PARMON
    - A comprehensive environment for monitoring large clusters
    - Use client-server techniques to provide transparent access to all nodes to be monitored
    - parmon-server & parmon-client

68

Need of more Computing Power: Grand Challenge Applications

Solving technology problems using computer *modeling*, *simulation* and *analysis*

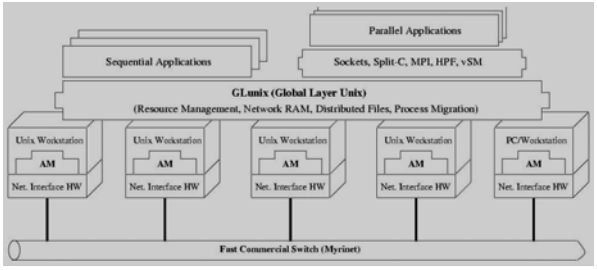
69

Representative Cluster Systems

- The Berkeley Network of Workstations (NOW) Project
  - Demonstrate building of a large-scale parallel computer system using mass produced commercial workstations & the latest commodity switch-based network components
  - Inter-Process Communication (IPC)
    - Active Messages (AM)
      - basic communication primitives in Berkeley NOW
      - A simplified remote procedure call that can be implemented efficiently on a wide range of hardware
  - Global Layer Unix (GLUnix)
    - An OS layer designed to provide transparent remote execution, support for interactive parallel & sequential jobs, load balancing, & backward compatibility for existing application binaries
    - Aim to provide a cluster-wide namespace and uses Network PIDs (NPIDs), and Virtual Node Numbers (VNNs)

70

Architecture of NOW System



71

Representative Cluster Systems

- Network RAM
  - Allow to utilize free resources on idle machines as a paging device for busy machines
  - Serverless
    - any machine can be a server when it is idle, or a client when it needs more memory than physically available
- xFS: Serverless Network File System
  - A serverless, distributed file system, which attempt to have low latency, high bandwidth access to file system data by distributing the functionality of the server among the clients
  - The function of locating data in xFS is distributed by having each client responsible for servicing requests on a subset of the files
  - File data is striped across multiple clients to provide high bandwidth

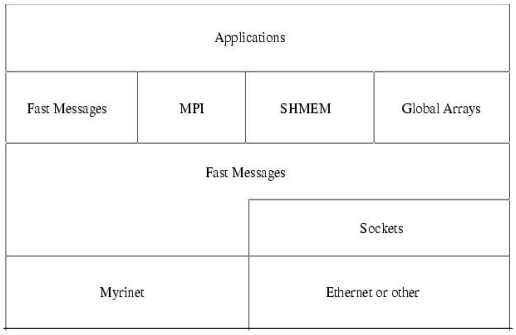
72

Representative Cluster Systems

- The High Performance Virtual Machine (HPVM) Project
  - Deliver supercomputer performance on a low cost COTS system
  - Hide the complexities of a distributed system behind a clean interface
  - Challenges addressed by HPVM
    - Delivering high performance communication to standard, high-level APIs
    - Coordinating scheduling and resource management
    - Managing heterogeneity

73

HPVM Layered Architecture



74

Representative Cluster Systems

- Fast Messages (FM)
  - A high bandwidth & low-latency comm protocol, based on Berkeley AM
  - Contains functions for sending long and short messages & for extracting messages from the network
  - Guarantees and controls the memory hierarchy
  - Guarantees reliable and ordered packet delivery as well as control over the scheduling of communication work
  - Originally developed on a Cray T3D & a cluster of SPARCstations connected by Myrinet hardware
  - Low-level software interface that delivery hardware communication performance
  - High-level layers interface offer greater functionality, application portability, and ease of use

75

Representative Cluster Systems

- The Beowulf Project
  - Investigate the potential of PC clusters for performing computational tasks
  - Refer to a PoPCs to describe a loose ensemble or cluster of PCs
  - Emphasize the use of mass-market commodity components, dedicated processors, and the use of a private communication network
  - Achieve the best overall system cost/performance ratio for the cluster

76

Representative Cluster Systems

- System Software
  - Grendel
    - the collection of software tools
    - resource management & support distributed applications
  - Communication
    - through TCP/IP over Ethernet internal to cluster
    - employ multiple Ethernet networks in parallel to satisfy the internal data transfer bandwidth required
    - achieved by 'channel binding' techniques
  - Extend the Linux kernel to allow a loose ensemble of nodes to participate in a number of global namespaces
  - Two Global Process ID (GPID) schemes
    - Independent of external libraries
    - GPID-PVM compatible with PVM Task ID format & uses PVM as its signal transport

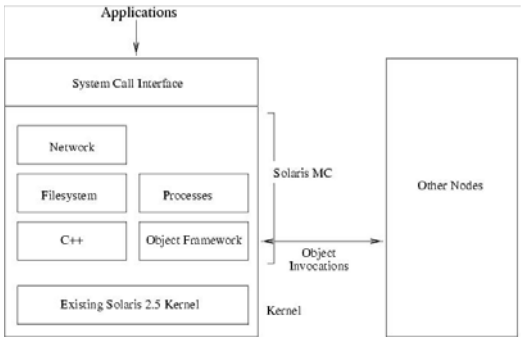
77

Representative Cluster Systems

- Solaris MC: A High Performance Operating System for Clusters
  - A distributed OS for a multicomputer, a cluster of computing nodes connected by a high-speed interconnect
  - Provide a single system image, making the cluster appear like a single machine to the user, to applications, and the the network
  - Built as a globalization layer on top of the existing Solaris kernel
  - Interesting features
    - extends existing Solaris OS
    - preserves the existing Solaris ABI/API compliance
    - provides support for high availability
    - uses C++, IDL, CORBA in the kernel
    - leverages spring technology

78

Solaris MC Architecture



79

Representative Cluster Systems

- Use an object-oriented framework for communication between nodes
  - Based on CORBA
  - Provide remote object method invocations
  - Provide object reference counting
  - Support multiple object handlers
- Single system image features
  - Global file system
    - Distributed file system, called ProXy File System (PXFS), provides a globalized file system without need for modifying the existing file system
  - Globalized process management
  - Globalized network and I/O

80

Cluster System Comparison Matrix

Project	Platform	Communications	OS	Other
Beowulf	PCs	Multiple Ethernet with TCP/IP	Linux and Grendel	MPI/PVM. Sockets and HPF
Berkeley NOW	Solaris-based PCs and workstations	Myrinet and Active Messages	Solaris + GLUnix + xFS	AM, PVM, MPI, HPF, Split-C
HPVM	PCs	Myrinet with Fast Messages	NT or Linux connection and global resource manager + LSF	Java-fronted, FM, Sockets, Global Arrays, SHEMEM and MPI
Solaris MC	Solaris-based PCs and workstations	Solaris-supported	Solaris + Globalization layer	C++ and CORBA

81

Cluster of SMPs (CLUMPS)

- Clusters of multiprocessors (CLUMPS)
  - To be the supercomputers of the future
  - Multiple SMPs with several network interfaces can be connected using high performance networks
  - 2 advantages
    - Benefit from the high performance, easy-to-use-and program SMP systems with a small number of CPUs
    - Clusters can be set up with moderate effort, resulting in easier administration and better support for data locality inside a node

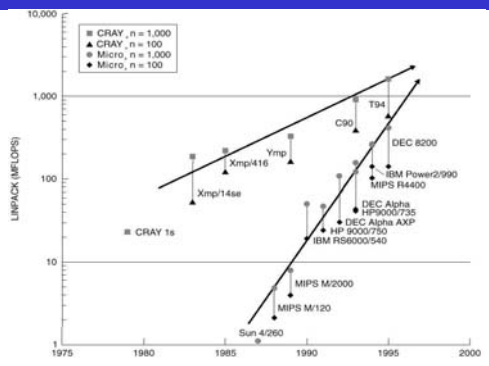
82

Hardware and Software Trends

- Network performance increase of tenfold using 100BaseT Ethernet with full duplex support
- The availability of switched network circuits, including full crossbar switches for proprietary network technologies such as Myrinet
- Workstation performance has improved significantly
- Improvement of microprocessor performance has led to the availability of desktop PCs with performance of low-end workstations at significant low cost
- Performance gap between supercomputer and commodity-based clusters is closing rapidly
- Parallel supercomputers are now equipped with COTS components, especially microprocessors
- Increasing usage of SMP nodes with two to four processors
- The average number of transistors on a chip is growing by about 40% per annum
- The clock frequency growth rate is about 30% per annum

83

Technology Trend



84

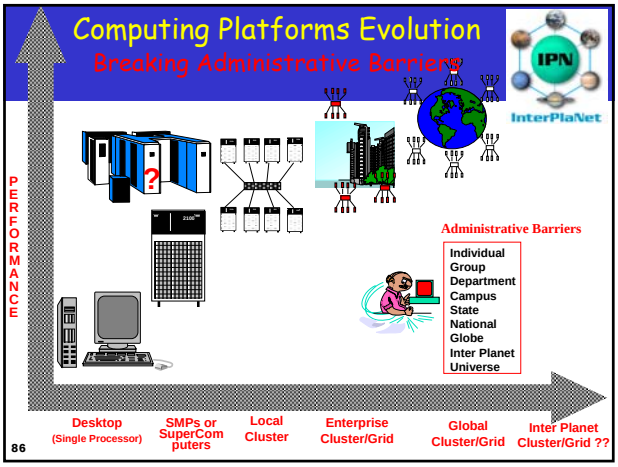
Advantages of using COTS-based Cluster Systems

- Price/performance when compared with a dedicated parallel supercomputer
- Incremental growth that often matches yearly funding patterns
- The provision of a multipurpose system

85

Computing Platforms Evolution

Breaking Administrative Barriers



86