Scalable Parallel Computing

Clustering for massive parallelism

- Computer cluster
  - Collection of interconnected stand-alone computers
    - Connected by a high-speed Ethernet connection
    - Work collectively and cooperatively as a single integrated computing resource pool
  - Massive parallelism at the job level
  - High availability through stand-alone operations
  - Scalable performance, high availability, fault tolerance, modular growth, COTS components
- Design objectives
  - Scalability
    - Modular growth; however, scaling from hundreds of uniprocessor nodes to 10,000+ multicore nodes is non-trivial
    - Scalability limited by factors including multicore chip technology, cluster topology, packaging method, power consumption, and cooling scheme applied
    - Other limiting factors include memory wall, disk I/O bottleneck, and latency tolerance
      - Memory wall is the growing disparity of speed between CPU and memory outside the CPU chip
  - Packaging
    - Nodes may be packaged as compact cluster or slack cluster
    - Affects communications wire length and interconnection techology
    - Compact cluster
      - Nodes closely packaged in one or more racks in a room
      - Nodes not attached to any peripherals
      - High bandwidth low latency communication network
    - Slack cluster
      - Nodes are complete SMPs, workstations, and PCs
      - May be located in different physical locations
      - Connected through standard LAN or WAN
  - Control
    - Centralized control for compact cluster
    - Centralized or decentralized control for slack cluster
    - Centralized control
      - All nodes owned, controlled, managed, and administered by a central operator
    - Decentralized control
      - Nodes owned individually
      - Owner may reconfigure, upgrade, or shut down the node at any time
      - Difficult to administer the complete system
      - May require special techniques for process scheduling, workload migration, checkpointing, and accounting
  - Homogeneity
    - Homogeneous cluster
      - All nodes are identical in terms of processor architecture and OS
      - Any node can execute binary code, even in mid-execution
    - Heterogeneous cluster

        - Nodes may be built on different platforms
        - Issues of interoperability (process migration for load balancing not possible)
  - Security
    - Exposed cluster
      - Communication paths among nodes are exposed to outside world
      - Individual nodes may be accessible using standard protocols, such as TCP/IP, and high overhead of those protocols
      - Not secure
      - Outside communications may disrupt intracluster communications in an unpredictable fashion
    - Enclosed cluster
      - Intracluster communications shielded from outside world
      - No standard protocol for efficient, enclosed intracluster communications
  - Dedicated vs enterprise clusters
    - Dedicated cluster
      - Typically installed in a deckside rack in a central computer room
      - Homogeneously configured with the same type of nodes
      - Managed by a single admin
      - Installed, used, and administered as a single machine
      - Much enhanced throughput and reduced response time
    - Enterprise cluster
      - Used to utilize idle resources in the nodes
      - Each node may be a full-fledged SMP, workstation, or PC
      - Nodes may be geographically distributed
      - Individual nodes owned and managed by different owners who may join or quit the cluster at any time
      - Cluster admin has limited control over the nodes
      - Owner's local jobs have priority over the enterprise jobs
      - Nodes connected through a low-cost Ethernet

- Fundamental cluster design issues
  - Scalable performance
    - Increase in performance by scaling of resources – cluster nodes, memory capacity, I/O bandwidth
    - Both scaling up and scaling down capabilities may be needed
  - Single-system image
    - A set of workstations connected together do not form a cluster
    - Combining several workstations into a megastation, with scaled performance
  - Availability support
    - High availability requirement
    - Redundancy in CPUs, memory, disks, I/O devices, networks, and OS images
  - Cluster job management
    - Goal to achieve high system utilization from nodes that may not be highly utilized otherwise
    - Job management software for batching, load balancing, and parallel processing
  - Internode communication
    - Not as compact as MPPs (massively parallel processors)
    - Longer wires increase latency
    - May also have issues with reliability, clock skew, and cross talking

    * Need reliable and secure communication protocols (such as TCP/IP) which increase overhead
  - Fault tolerance and recovery
    * Can be designed to eliminate all single points of failure
    * In case of node failure, critical jobs running on failing nodes can be saved to the surviving nodes
    * Use rollback with periodic checkpointing
  - Cluster family classification
    1. Compute clusters
       * Beowulf clusters
       * Designed for collective computing over a single large job
       * Numerical simulation of weather conditions
       * Do not handle many I/O operations
       * Dedicated network to facilitate communication among cluster nodes, with homogeneous nodes
       * Use message passing interface (MPI) or parallel virtual machine (PVM) for porting the code
    2. High availability clusters
       * Designed for fault tolerance and high availability
       * Multiple redundant nodes to sustain faults or failures
    3. Load-balancing clusters
       * Aim for higher resource utilization through load balancing
       * All nodes share the workload as a single VM
       * Requests initiated by the user are distributed to all nodes
       * Need middleware to achieve dynamic load balancing by job or process migration among all cluster nodes

Computer Clusters and MPP Architectures

- Cluster organization and resource sharing
  - Basic cluster architecture
    * Simple cluster can be built with commodity components
    * Commodity workstations as cluster nodes
    * Nodes interconnected by a fast commodity network, using standard communication protocols
    * Deploy cluster middleware to glue together all node platforms in user space, offering HA service
    * Use an SSI layer to provide a single entry point, a single file hierarchy, a single point of control, and a single job management system
    * Idealized cluster supported by three subsystems
      1. Conventional databases and online transaction processing (OLTP) monitors offer users a desktop environment to use the cluster
      2. In addition to running sequential user programs, cluster supports parallel programming based on standard languages and clustering libraries using PVM, MPI, or OpenMP
      3. A user interface subsystem to combine the advantages of web interface and Windows GUI
  - Resource sharing in clusters