Cluster Computing

Dependable Clustered Computing

Introduction

- Two different areas of computing with slightly different motivations
  - High performance (HP)
  - High availability (HA)
- The clustering model can provide both HA & HP, and also manageability, scalability, & affordability
  - To accomplish the dream, cluster software technologies still have to evolve & mature
- Clusters
  - Poor man's answer for the parallel high performance computing quest
    - Observed in Gordon Bell prize
  - Typically homogeneous, tightly coupled, nodes trust each other
    - Compare with distributed systems
  - As number of h/w components rises, so does the probability of failure
    - (Leslie Lamport) " a distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable"
    - Increasing probability of fault occurrence for long-running applications

2

Failover

- In case of failure of a component in the primary system, the critical applications can be transferred to the secondary server, thus avoiding downtime and guaranteeing application's availability
- Inevitably imply a temporary degradation of the service, but without total permanent loss
- Fault-tolerant servers, pioneered by Tandem

3

Two Worlds Converge

- Only fault tolerance can enable effective parallel computing by allowing application continuity
- Parallelism enables high-availability by providing redundancy

4

Dependable Parallel Computing

- Dependability
  - Arose as an issue as soon as clusters started to be used by HP community
- Two main reasons
  - As systems scale up to hundreds of nodes, the likelihood of a node failure increases proportionally with number of nodes
    - caused not only by permanent faults (permanent damage), but mainly by transient ones (due to heating, electromagnetic interference, marginal design, transient SW faults, and power supply disturbances)
  - Those faults that statistically don't cause node failures, but have other insidious effects
    - the effect of faults on application running in parallel systems
      - a large percentage of faults do not cause nodes to crash, system panics, core dumps, or application to hang
      - in many cases, everything goes apparently fine, but in the end, parallel application generate incorrect results

5

Dependable Parallel Computing

- Examples
  - ASCI Red : 9000 Intel Pentium Pro (a 263 Gbytes memory machine)
    - MTBF : 10 hours for permanent faults, 1-2 hours for transient faults (a node's MTBF is 10 years)
  - ORNL (MPP, not cluster)
    - MTBF is 20 hours
  - Carnegie Mellon University (400 nodes)
    - MTBF is 14 hours

6

Mission/Business Critical Computing

- The demand for HA applications increases
  - Electronic commerce, WWW, data warehouse, OLAP, etc.
  - the cost of outage is substantial
- Development of dependable computing
  - Cold-backup
    - one of the first protection mechanism for critical applications
    - data offline backup
  - Hot-backup
    - Protect data, not just at a certain time of the day, but on a continuous basis using a mirrored storage system where all data is replicated
    - RAID were introduced to protect data integrity
- Advanced H/W redundancy system
  - Redundancy at all system level to eliminate single point of failure
  - Fault tolerant, 99.999 % availability (5 mins downtime per year)

7

Mission/Business Critical Computing

- Combine highly reliable & highly scalable computing solutions with open & affordable components
  - Can be done with clusters
  - The ability to scale to meet the demands of service growth became a requirement
  - High performance joined high-availability as an additional requirement for mission/business critical systems
- Clusters intrinsically have the ability to provide HP & scalability
  - High availability scalable clusters
- Today's mission/business critical clusters as a combination of the following capabilities
  - Availability: one system to failover to another system
  - Scalability: increase the number of nodes to handle load increase
  - Performance: perform workload distribution
  - Manageability: manage a single system
    - as a requirement
    - clusterware

8

Dependability Concepts

- Faults, Errors, Failures
  - Error detection latency
    - The time between the first occurrence of the error and the moment when it is detected
    - The longer, the most affected the internal state of the system
  - Fault
    - When an error has some phenomenological cause
    - Transient
      - Re-execution can compensate for a transient fault
    - Permanent
      - More complex to handle
      - Software faults (bugs)
        - Bohrbugs
          – bugs are always there again when the program is re-executed
        - Heisenbugs
          - in a re-execution, the timing will usually be slightly different, the bug does not happen again

9

Dependability Concepts

- Dependability Attributes
  - Reliability
    - Probability of that continuous correct operation
  - Availability
    - A measure of the probability that the system will be providing good service at a given point in time
  - MTBF - The Mean Time Between Failures
  - MTTR - The Mean Time To Repair a failure, & bring the system back to correct service
  - Availability can be expressed as
    MTBF / (MTBF+MTTR)

10

Dependability Concepts

- Dependability Means
  - Two ways
    - Fault prevention
      - Preventing faults from occurring
      - Accomplished through conservative design, formal proofs, extensive testing, etc
    - Fault tolerance
      - Prevent errors from becoming failures
  - 2 approaches are complementary
  - Disaster Tolerance
    - Techniques aimed at handling a very particular type of faults, namely site failures caused by fire, floods, sabotages, etc
    - Usually implies having redundant hardware at a remote cite

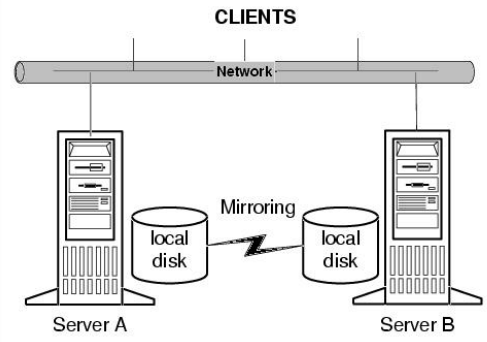
11

Cluster Architectures

- Share-Nothing versus Shared-Storage
  - Share-nothing cluster model
    - Each node has its own memory & own storage resources
    - Allow nodes to access common devices or resources, as long as these resources are owned and managed by a single system at a time
    - Avoid the complexity of cache coherency schemes & distributed lock managers
    - May eliminate single point of failure within the cluster storage
    - Scalable, because of the lack of contention & overhead
      - Several strategies aimed at reducing bandwidth overhead
  - Shared-Storage
    - Both servers share the same storage resource with the consequent need for synchronization between disks accesses to keep data integrity
    - Distributed lock managers must be used
    - Scalability becomes compromised

12

Shared-nothing Clusters



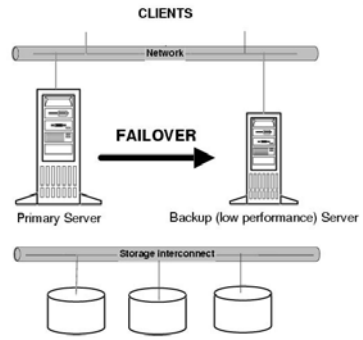
13

Cluster Architectures

- Active/Standby versus Active/Active
  - Active/Standby
    - Called hot backup
    - Primary server where the critical application runs
    - Secondary server (hot spare)
      - normally in standby mode
      - can be a lower performance machine
  - Active/Active
    - All servers are active & do not sit idle waiting for a fault to occur & take over
    - Providing bidirectional-failover
    - The price to performance ratio decrease
  - N-Way
    - Several active servers that back up one another

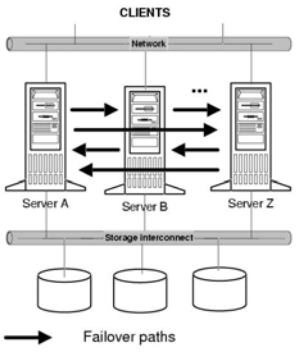
14

Active/Standby Clusters



15

N-Way Clusters



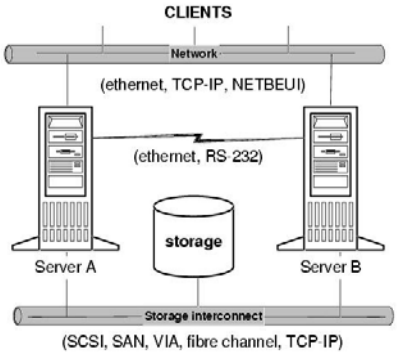
16

Cluster Architectures

- Interconnects
  - 3 main kinds of interconnects – network, storage, monitoring
    - Vital for clusters
  - Network interconnect
    - Communication channels that provides the network connection between the client systems & the cluster
    - Generally Ethernet or fast-Ethernet
  - Storage
    - Provide the I/O connection between the cluster nodes & disk storage
    - SCSI and Fiber Channel
      - SCSI is more popular, limit in length (SCSI extender technologies)
      - FC-AL permits fast connections - 100Mb/s - & up to 10 Km
  - Monitoring
    - Additional comm media used for monitoring purposes (heartbeats transfer)
    - NCR Lifekeeper
      - use both the network & storage interconnect for monitoring
      - also have an additional serial link between the servers for that purpose

17

Cluster Interconnects



18

Cluster Architectures

- VIA
  - Virtual Interface (VI) Architecture
  - An open industry specification promoted by Intel, Compaq, & Microsoft
  - Aim at defining the interface for high performance interconnection of servers and storage devices within a cluster (SAN)
  - VI proposes to standardize the network (SAN)
  - Define a thin, fast interface that connects software applications directly to the networking hardware while retaining the security & protection of the OS
  - VI-SAN
    - Specially optimized for high bandwidth & low latency comm

19

Cluster Architectures

- Storage Area Network & Fiber Channel
  - SAN
    - To eliminate the bandwidth bottlenecks & scalability limitations imposed by previous storage (mainly SCSI bus-based) architectures
    - SCSI is based on the concept of a single host transferring data to a single LUN (logical unit) in a serial manner
  - FC-AL
    - Emerged as the high-speed, serial technology of choice for server-storage connectivity
    - The most widely endorsed open standard for the SAN environment
    - High bandwidth & scalability,
    - Support multiple protocols (SCSI, IP) over a single physical connection
  - Modular scalability
    - Most important advantages of the SAN approach
    - A key to enabling infrastructure for long term growth & manageability

20

Detecting and Masking Faults

- A dependable systems is built upon several layers
  - EDM (Error Detection Mechanism)
    - Classified according to 2 important characteristics
      - Error coverage
        - The percentage of errors that are detected
      - Detection latency
        - The time an error takes to be detected
  - Diagnosis layer
    - Collect as much info as possible about the location, source & affected parts
    - Run diagnosis over the failed system to determine whether the fault is transient or permanent
  - Recovery & reconfiguration

21

Detecting and Masking Faults

- Self-Testing
  - To detect errors generated by permanent faults
  - Consists of executing specific programs that exercise different parts of the system, & validating the output to the expected known results
  - Not very effective in detecting transient faults
- Processor, Memory, and Buses
  - Parity bits - primary memories
  - Parity checking circuitry – system buses
  - EDAC (error detection & correction) chips
    - common in workstation hardware
  - Built-in error detection mechanisms
    - inside of microprocessors
  - Fail to detect problem occurs at a higher level or affects data in subtle ways

22

Detecting and Masking Faults

- Watchdog Hardware Timers
  - Provide a simple way of keeping track of proper process functions
  - If the timer is not reset before it expires, the process has probably failed in some way
  - Provide an indication of possible process failure
  - Coverage is limited
    - data & results are not checked
  - Implemented in SW or HW
- Loosing the Software Watchdog
  - Software watchdog
    - A process that monitors other process(es) for errors
    - (simplest) Watch its application until it eventually crashes, the only action it takes is to relaunch the applications
  - Monitored process is instrumented to cooperate with the watchdog
  - WinFT
    - Typical example of a SW watchdog

23

Detecting and Masking Faults

- Heartbeats
  - A periodic notification sent by the application to the watchdog to assert its aliveness
  - Consist of an application–initiated "I'm Alive" message, or a request-response scheme (watchdog requests the application to assert its aliveness through "Are you Alive?" message) & wait for acknowledgement
  - Can coexist in the same system
    - Useless if OS crashes
  - Clustering solutions provide several alternate heartbeat paths
    - TCP-IP, RS-232, a shared SCSI bus
- Idle notification
  - Inform the watchdog about periods when the application is idle, or it is not doing any useful work, then the watchdog can perform preventive actions (restarting application)
  - Software Rejuvenation
    - software gets older, the probability of faults is higher
- Error notification
  - An application that is encountering errors which it can't overcome can signal the problem to the watchdog & request recovery actions
  - Restart may be enough

24

Detecting and Masking Faults

- Assertions, Consistency Checking, and ABFT
  - Consistency checking
    - Simple fault detection technique that can be implemented both at the hardware and at the programming level
    - Performed by verifying that the intermediate or final results of some operation are reasonable
    - At the hardware level, built-in consistency checks for checking addresses, opcodes, and arithmetic operations
  - Range check
    - Confirm that a computed value is within a valid range
  - Algorithm-Based Fault Tolerance (ABFT)
    - After executing an algorithm, the application runs a consistency check specific for that algorithm
    - Checksum for matrix operations
      - Extend the matrices with additional columns

25

Recovering from Faults

- Checkpointing and Rollback
  - Checkpointing
    - Allow a process to save its state at regular intervals during normal execution so that it can be restored later after a failure to reduce the amount of lost work
    - When a fault occurs, the affected process can simply be restarted from the last saved checkpoint (state) rather than from the beginning
    - Protect long-running applications against transient faults
    - Suitable to recover applications from software errors
  - Checkpointing classifications
    - Transparency (data to include in the checkpoint)
      - Checkpoints can be transparently or automatically inserted at runtime (or by the compiler)
        - Save large amounts of data
      - Inserted manually by the application programmer
    - The checkpointing interval
      - The time interval between consecutive checkpoints
      - Critical for automatic checkpointing

26

Recovering from Faults

- Transactions
  - A group of operations which form a consistent transformation of state
    - operations: DB updates, messages, or external actions
  - ACID properties
    - Atomicity
      - Either all or none of the actions should happen commit or abort
    - Consistency
      - Each transformation should see a correct picture of the state, even if concurrent transactions are updating the state
    - Integrity
      - The transaction should be a correct state transformation
    - Durability
      - Once a transaction commit, all its effects must be preserved, even if there is a failure

27

Recovering from Faults

- Failover and Failback
  - Failover process
    - The core of the cluster recovery model
    - A situation in which the failure of one node causes a switch to an alternative or backup node
    - Should be totally automatic and transparent, without the need for administrator intervention or client manual reconnection
    - Require a number of resources to be switched over to the new system (switch network identity)
  - Takeover
    - Used in fault-tolerant systems
    - Built with multiple components running in lock-step
  - Failback
    - Move back the applications/clients to the original server once it is repaired
    - Should be automatic & transparent
    - Used efficiently for another purpose: maintenance

28

The Practice of Dependable Clustered Computing

- MS Cluster Server
  - Wolfpack
    - Provide HA over legacy windows applications, as well as provide tools & mechanisms to create cluster-aware applications
    - Inherited much of the experience of Tandem in HA
      - Tandem provided the Nonstop SW & the ServerNet from Himalaya Servers
  - A phased approach
    - Phase I: support 2 node clusters & failover of one node to another
    - Phase II: a multinode share-nothing architecture scalable up to 16 nodes

29

The Practice of Dependable Clustered Computing

- NCR LifeKeeper
  - In 1980 NCR began delivering failover products to the telecomm industry
  - LifeKeeper in the Unix world in beginning of 90s
  - Recently ported to Windows NT
  - Multiple fault detection mechanisms
    - Heartbeat via SCSI, LAN, & RS 232, event monitoring, & Log monitoring
  - 3 different configurations
    - Active/active, active/standby, & N-way (with N=3)
  - Recovery action SW kits for core system components
  - Provide a single point of administration & remote administration capabilities

30

The Practice of Dependable Clustered Computing

- Oracle Failsafe and Parallel Server
  - Oracle DB options for clustered systems
  - The archetype of a cluster solution specially designed for DB systems
  - Oracle Failsafe
    - Based on MSCS
    - Support a max of 2 nodes (due to limitations of MSCS phase I)
    - Targeted for workgroup and department servers
  - Oracle Parallel Server
    - Support a single DB running across multiple nodes in a cluster
    - Not require MSCS
    - Support more than 2 nodes
    - Targeted for enterprise level systems, seeking not only HA but also scalability
    - Support Digital Clusters for Windows NT & IBM's "Phoenix" technology
  - Oracle servers
    - Support hot-backups, replication, & mirrored solutions

31