# Server Architectures:
## Cluster and Massively Parallel Machines

*January 2005*

**René J. Chevance**

## Foreword

- **This presentation is an introduction to a set of presentations about server architectures.**
  **They are based on the following book:**

  **Serveurs Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, Storage Solutions**
  **René J. Chevance**
  **Digital Press December 2004 ISBN 1-55558-333-4**
  **http://books.elsevier.com/**

  **This book has been derived from the following one:**

  **Serveurs multiprocesseurs, clusters et architectures parallèles**
  **René J. Chevance**
  **Eyrolles Avril 2000 ISBN 2-212-09114-1**
  **http://www.eyrolles.com/**

  **The English version integrates a lot of updates as well as a new chapter on Storage Solutions.**





**Contact: www.chevance.com** **rjc@chevance.com**

# Organization of the Presentations

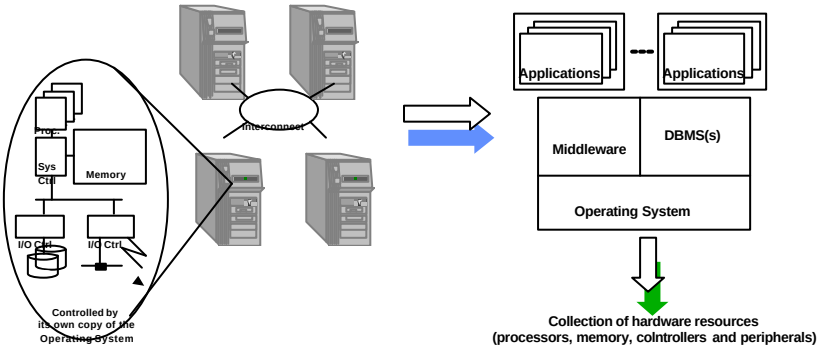
- Introduction
- Processors and Memories
- Input/Output
- Evolution of Software Technologies
- Symmetric Multi-Processors
- ➔ Cluster and Massively Parallel Machines (this document)
  - Cluster and MPP Architecture Overview
  - Clusters
    - IBM Cluster 1600
    - Microsoft Cluster Service
  - Shared Memory Clusters
    - IBM Sysplex
    - SGI Altix 3000
  - Advantages and disadvantages of Cluster architecture
  - Massively Parallel Processing – MPP
    - IBM Cluster 1600 for High Performance Computing
  - Advantages and disadvantages of MPP architecture
  - Grid Computing
  - Global Computing (GC) and Peer-to-Peer (P2P)
  - SMPs, Clusters and MPPs - a Summary
  - Flexible SMP-Cluster Architecture
  - Positioning of the various options of architecture
  - Flexible SMP-Cluster Architecture
    - Unisys' flexible ES7000 architecture
  - Vector Architecture Example: the NEC SX-6
- Data Storage
- System Performance and Estimation Techniques
- DBMS and Server Architectures
- High Availability Systems
- Selection Criteria and Total Cost of Possession
- Conclusion and Prospects





# Cluster and MPP Architecture Overview

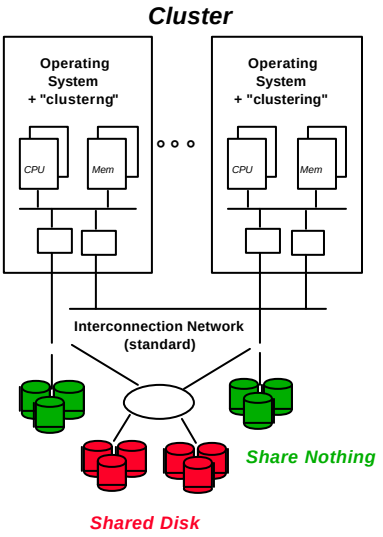
## Loosely Coupled Architecture Overview



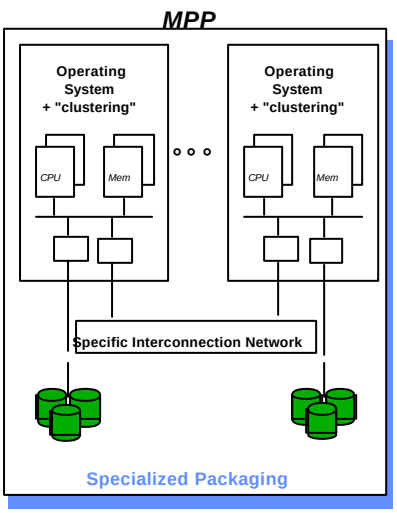

a) Loosely-coupled Architecture

b) Idealized View of a Loosely-Coupled Architecture

## Cluster and MPP Architecture Overview(2)




*Two architectural options for connection to disks*

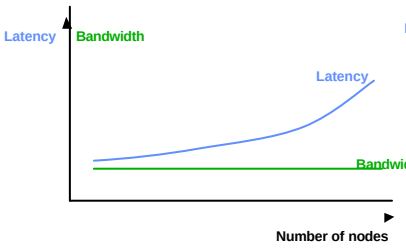



*Typically, only one architectural option for connection to disks*

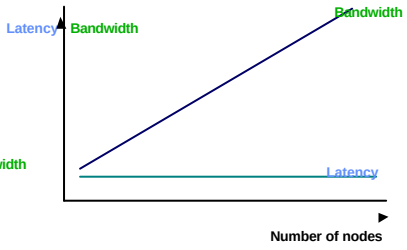



## Cluster and MPP Architecture Overview(3)

- **Characteristics of interconnection networks**




a) Typical Cluster




b) Ideal case MPP

## Clusters

- Set of interconnected nodes
- Standard technology for the interconnection network (e.g. fast Ethernet, Fibre Channel)
- Each node has its own set of resources (processor(s), memory, I/Os) and is operating under the control of its own copy of the Operating System
- It is a loose coupling since systems are not sharing memory (general case)
- Differences with distributed systems: homogeneity of nodes (vendor, operating system), geographic proximity, and single system image for given resources
- Communication between applications running on different nodes is through message passing
- Operating system (usually) derived from a non-cluster OS with:
  - Single System Image (SSI) for system administrator
  - Users have a transparent access to certain resources (said as clusterized resources) such as File Systems, unique IP address for the cluster,…
- Concept introduced in the late 70's by Tandem (objective was Fault Tolerance) and made popular by DEC starting in 1983 (objective was scalability)




## Clusters(2)

- **Unix Clusters**
  - Many offers from different vendors with similar functionalities but different implementation (specific to each vendor)
  - Usually, robust solutions
  - Clusterized File Systems were introduced quite late
  - Lack of standard for API and SDK
  - Example of an attempt to provide a portable cluster solution: Veritas High Availability or High Performance Computing
- **Linux world: many solutions aimed at either High Availability or High Performance Computing**

## Clusters(3)

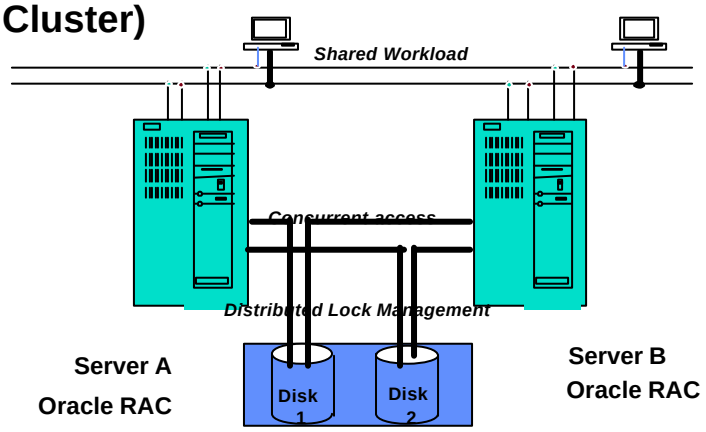
- **IBM Cluster 1600**
  - Convergence between HACMP (High Availability Cluster MultiProcessing) and RS/6000 SP (a MPP) offers
  - Two flavors:
    - Cluster 1600 for Commercial Computing
    - Cluster 1600 for High Performance Computing
  - Components:
    - CSM (Cluster System Management) or PSSP (Parallel System Support Program) for system management
    - GPFS (General Parallel File System), a clusterized File System
    - Parallel Environment for the construction of distributed applications which is composed of:
      - Communication Subsystem: support of SP Switch and SP Switch 2 interconnection networks;
      - Virtual Shared Disk (VSD): creation of logical volumes accessible from any node (that is, construction of a Shared Disk architecture on top of a Shared Nothing architecture);
      - Recoverable Virtual Shared Disk (RVSD): allowing continuity of accessibility in the event that a node fails





## Clusters(4)

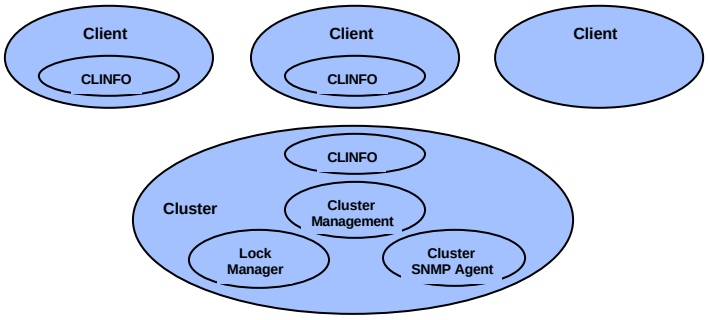
- **Example of a configuration of Cluster 1600 using Oracle RAC (Real Application Cluster)**




*Note: Conifuration is not limited to two nodes*

## Clusters(5)

- **System Architecture (HACMP – after IBM)**




- **More details on components and functionalities within the « High Availability Systems » presentation**

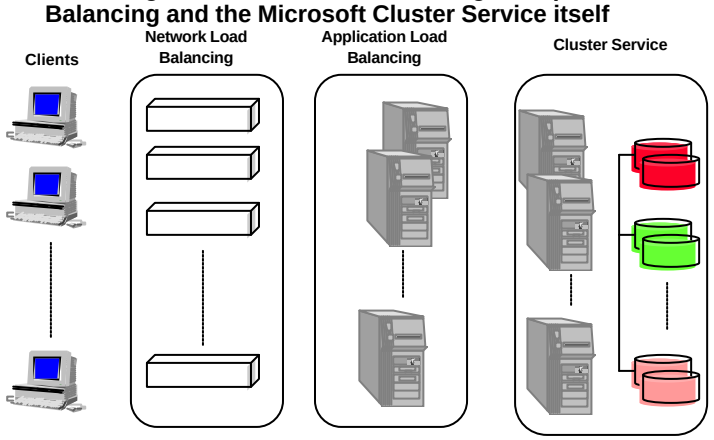



## Clusters(6)

- **Components:**
  - **The Cluster Manager: keeping up to date information on the cluster as well as the state of the interconnect network**
  - **CLINFO. The Cluster Information Services: allows clients to query the state of the cluster**
  - **Cluster SNMP Agent**
  - **Lock Manager: provides a distributed synchronization service to the various services and applications executing on the cluster**

## Clusters(7)

- **Microsoft Cluster Service (after Microsoft)**
  - **Example of use of Windows Cluster using the three technologies: Network Load Balancing, Component Load Balancing and the Microsoft Cluster Service itself**

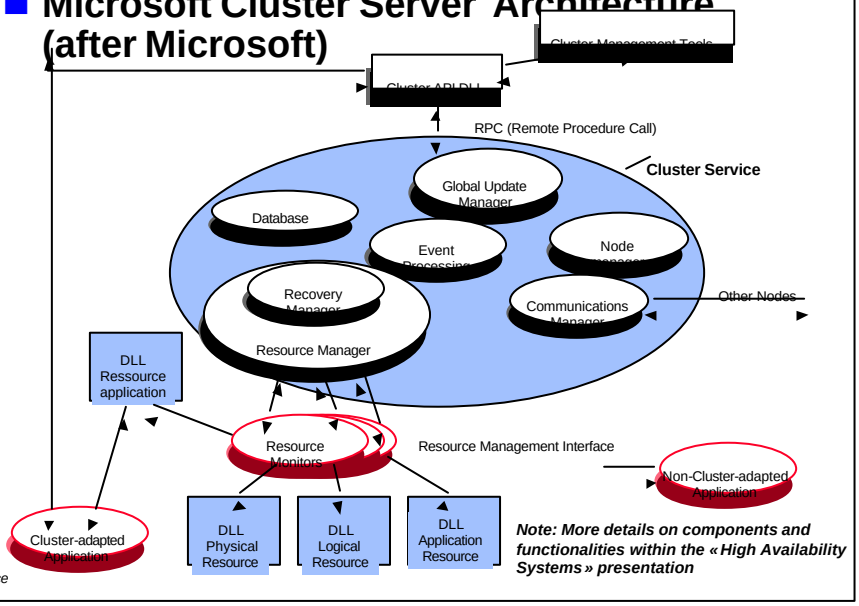






## Clusters(8)

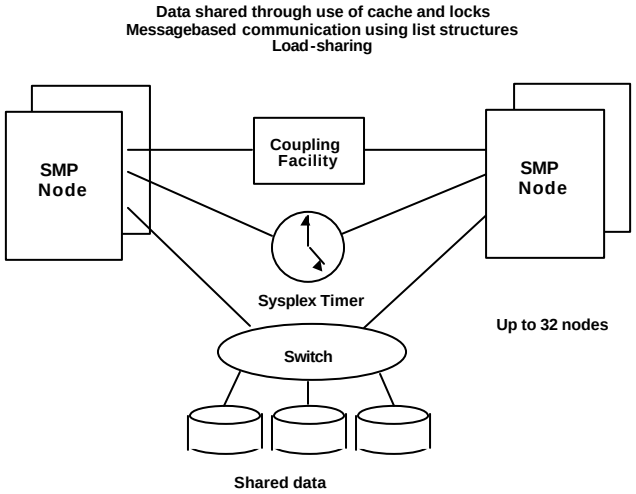
- **Microsoft Cluster Server Architecture (after Microsoft)**




***Note: More details on components and functionalities within the « High Availability Systems » presentation***

## Shared Memory Clusters

- **Logical Architecture of IBM Sysplex (after IBM)**








## Shared Memory Clusters(2)

- **Components:**
  - **Coupling Facility: Shared data cache between nodes (with coherency) and control structures providing shared data queues and distributed locking**
  - **Sysplex Timer: providing time base to the nodes**
  - **Switch: connection to disks (ESCON or Fibre Channel)**
- **Two types of Sysplex Cluster:**
  - **Local cluster (proximity of nodes)**
  - **GDPS (Geographically-Dispersed Cluster) which provides Disaster Recovery in conjunction with data duplication techniques (PPRC Peer-to-Peer Remote Copy or XRC eXtended Remote Copy, see « Data Storage » presentation)**
    - **GDPS/PPRC: less than one hour recovery time and a maximum of 40 kilometers between sites**
    - **GDPS/XRC: about 30 minutes of recovery time and no distance limitation**

## Shared Memory Clusters(3)

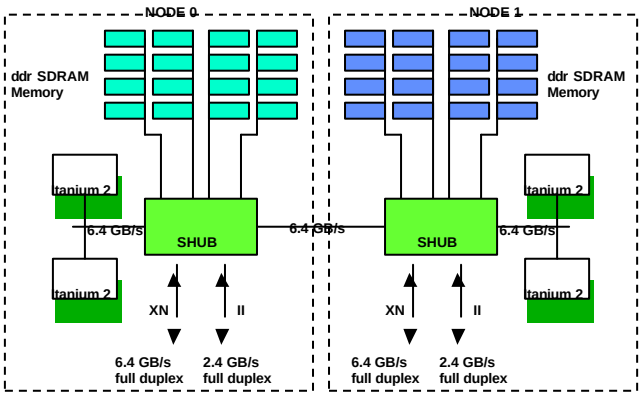
- **Resouce Management:**
  - **PR/SM (Processor Resource/System Manager)**
  - **WLM (Work Load Manager)**
- **Products taking benefit of the Coupling Facility:**
  - **Batch Processing**
  - **Data Management: DB2, IMS/DB and VSAM RLS (Record Level Sharing)**
  - **Transaction Monitors: CISC and IMS/TM**
  - **WLM**
  - **WebSphere**
- ***Note: Shared Memory and High Availability***
  - **Following the failure of one node in a cluster, the question of the state of the shared memory is central. In the general case, it is not possible to recover a consistent state for the shared memory nor to insure that its current state is correct.**
    **Author's hypothesis: as only « proprietary » software products have access to the shared memory, one easy way to insure consistency and recoverability is to structure any update to the shared memory as a transaction (ACID property).**





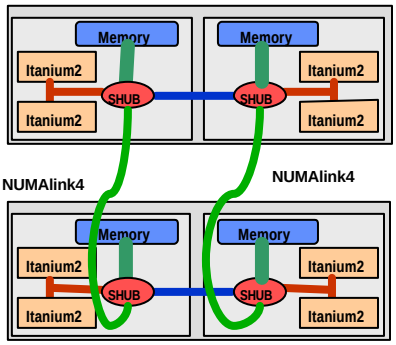
## Shared Memory Clusters(4)

- **Newest version of the Origin 3000 based upon Itanium**
- **Architecture of the Altix 3000 Itanium-2 based C-brick (according to SGI)**

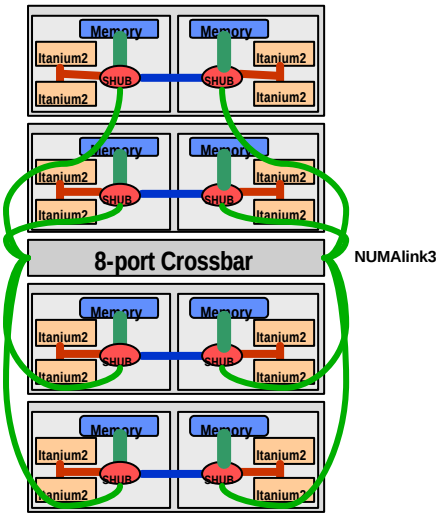


## Shared Memory Clusters(5)

- **Some examples of configurations Altix 3000 (Source SGI)**




a) 8-processor Configuration without router



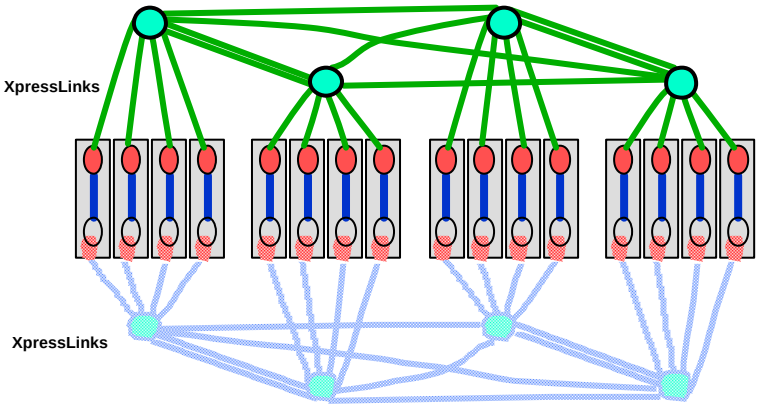

b) 16-processor configuration with router





## Shared Memory Clusters(6)

- **Some examples of configurations Altix 3000 (Source SGI)**




c) 64-processor Configuration

## Shared Memory Clusters(7)

- **Altix 3000 can be viewed as either:**
  - **a CC-NUMA SMP**
  - **an SMP-based cluster, making use of CC-NUMA for any node larger than 4 processors, with shared memory**
- **Altix is using fat tree topology**
- **Linux has been extended to cope with CC-NUMA (affinity)**
- **The standard MPI (Message Passing Interface) and SHMEM (Shared memory) interfaces have been implemented to take benefit of the BTE hardware (Block Transfer Engine) which provides fast data movement and synchronization operations)**




## Advantages and disadvantages of Cluster architecture

- **Advantages and Disadvantages of the Cluster Approach**

| Advantages | Disadvantages |
|---|---|
| High intrinsic availability (independence of the nodes); | Multiprocessor effectiveness limited (compared to SMP) |
| Simple hardware implementation; | Implies the need for changes to the OS (making Single System Image more difficult); |
| Application-level compatibility with uniprocessors and multipro cessors; | Applications must be modified to take advantage of cluster performance increase (i.e. to give the application the ability to exploit several nodes concurrently). |
| Increase in performance for DBMS (OLTP if there are few interactions between nodes, Decision Support always); | In practice, for commercial computing, the only software ported to clusters has been DBMS's (e.g. Oracle, DB2, SQL Server); |
| Easy integration of new technologies (processors and OS versions); | The standards for writing parallel programs are still in their early stages; |
| Transparent sharing of "clusterized" resources; | The upper limit to the size of a cluster is only of the order of ten interconnected systems; |
| Ease of maintenance. | Difficulties administering the cluster. |

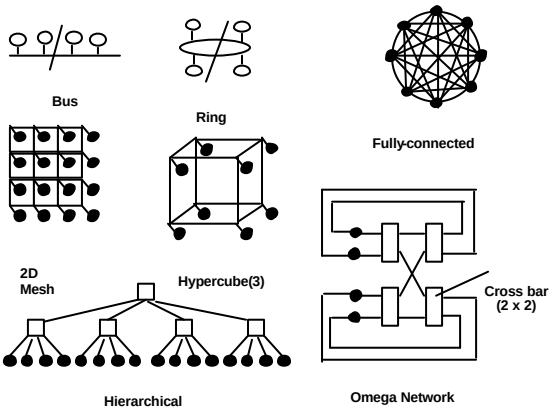
## Massively Parallel Processing - MPP

- Set of interconnected nodes through a specialized network (low latency, high throughput and scalability)
- Specialized packaging aimed at supporting a very large number of nodes
- Very large number of nodes: O(100) up to O(1000)
- Each node run under the control of its own copy of the Operating System (general case)
- Objective: Search for very high application performance by exploiting parallelism
- In practice, few applications have been parallelized: DBMSes and High Performance Computing
- First MPPs were based upon uni-processor nodes. SMP nodes are common place
- Few offers remaining




## Massively Parallel Processing – MPP(2)

- **Examples of interconenct technologies**



## Massively Parallel Processing – MPP(3)

- **A comparison between several interconnect networks used in a 64- node system (source [PAT04])**

| Criterion | | Bus | Ring | 2D Mesh | Hypercube (6) | Fully connected |
|---|---|---|---|---|---|---|
| Bandwidth | Total bandwidth | 1 | 64 | 112 | 192 | 2 016 |
| | Bisection BW | 1 | 2 | 8 | 32 | 1 024 |
| Cost | Port by switch | N/A | 3 | 5 | 7 | 64 |
| | Total number of links | 1 | 128 | 176 | 256 | 2 080 |

- **Characteristics of SGI's Spider interconnect network (source [GAL97])**

| Number of nodes | Average latency (NS) | Bisection BW (GB/s) |
|---|---|---|
| 8 | 118 | 6.4 |
| 16 | 156 | 12.8 |
| 64 | 274 | 51.2 |
| 256 | 344 | 205.0 |
| 512 | 371 | 410.0 |




## Massively Parallel Processing – MPP(4)

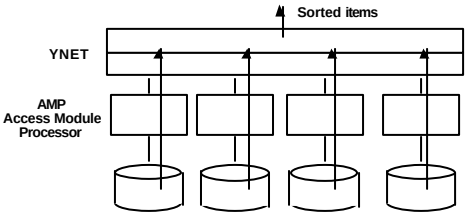
- **IBM Cluster 1600 for High Performance Computing**
  - **Interconnect: either Ethernet or SP Swich (SP Switch2 or an earlier generation of the SP Switch)**
  - **SMP nodes based upon Power processors.**
  - **From 16 to 128 nodes depending on node type (Thin, Wide or High)**
- **Switch2 Performance (Source [IBM03])**

| Number of nodes | Latency (µsec) | Bandwidth (MB/s) | |
|---|---|---|---|
| | | One-way | Bidirectional |
| Up to 16 | 1.0 | 500 | 1000 |
| From 17 to 80 | 1.5 | | |
| From 81 to 512 | 2.5 | | |

| Number of TCP/IP sessions | Bandwidth (MB/s) Single | | Bandwidth (MB/s) Dual | |
|---|---|---|---|---|
| | Unidirectional | Bidirectional | Unidirectional | Bidirectional |
| 1 or more | 238 | 259 | - | - |
| 1 | - | - | 449 | 495 |
| 2 or more | - | - | 435 | 491 |

## Massively Parallel Processing – MPP(5)

- **NCR WorldMark 5250**
  - **Successor of the system introduced in 1984: system connected to mainframes for decision support applications**
  - **System based upon 8086 and proprietary OS and DBMS**
- **Operation of the first Teradata system**
  - **Requests were broken into sub-requests and submitted to AMPs (Access Module Processors)**
  - **Selected items are presented (AMPs) to the YNET which performed a sort operation**

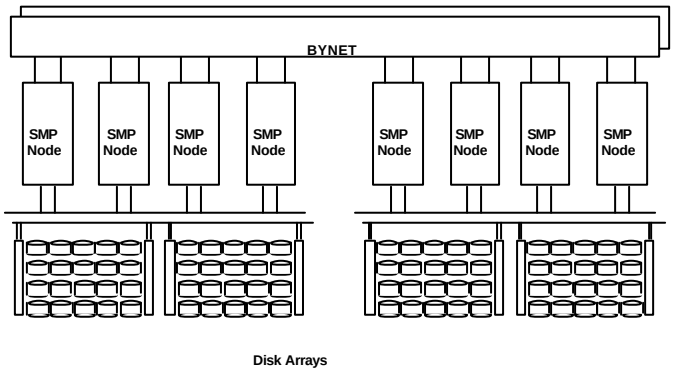






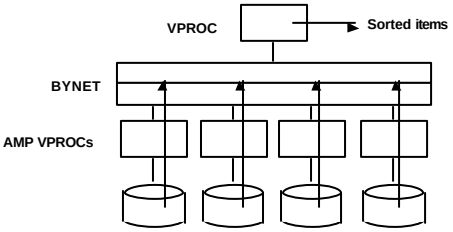
## Massively Parallel Processing – MPP(6)

- **General architecture of the NCR WorldMark 5250 (Source [SWE02])**
  - **Based on Intel-based 4_way SMP nodes**
  - **BYNET: redundant network based upon 64 way switches (120 MB/s per BYNET connection)**



# Massively Parallel Processing – MPP(7)

- **NCR WorldMark 5250: Concept of VPROC**
  - **At the time of the first Teradata machines, the available processors (Intel 8086) were slow, and the memories of low capacity. It therefore made sense to distribute processing between processors and the network.With the march of technology, processors became very fast and memories became very large making a simple, highly-scalable interconnect network more attractive and relying on fast processors and large memories in the nodes to do the sorting. BYNET is omiting sorting capability.**
  - **Requests are still broken in sub-requests, results are passed to the BYNET for sending to the consuming processes**
  - **VPROC (Virtual Processes) is the abstraction supporting the operations (e.g. tuple selection, sorting,...)**








# Advantages and disadvantages of MPP architecture

- **A summary of the advantages and disadvantages of MPP**

| Advantages | Disadvantages |
|---|---|
| Favorable cost/performance ratio compared with traditional vector supercomputers for scientific applications | Area of effectiveness is limited |
| Performance scalability limited only by the degree of parallelization of the application (that is, the MPP does not have the size limits (in terms of number of processors) that clusters and SMPs have) | Non-standard specialized interconnect networks |
| | An emergent technology, not a mature one |
| Potential of high performance in decision support applications (given an adapted DBMS) | Implies modifications of the system (e.g. SSI) |
| | Emerging programming standards for parallel applications |
| High availability (potentially) | Limited number of applications available |
| | Difficulty of writing parallel applications |
| | Difficulties in administering the system |

- **Note: From a system architecture perspective, clusters and MPPs are similar (e.g. same OS adaptations, same clustering software,....) but their optimization points are different: clusters are tuned for High Availability and MPPs are tuned at High Performance Computing**

## Grid Computing

- During the 80s, it was observed that workstations were inactive most of the time. So, comes the idea of making use of « idle » cycles. This is the concept of No use NOW (Network Of Workstations) interconnected by a LAN or a specific technology
- Bandwidth and Latency of NOW interconnect networks

| Type of network | Flow | Latency |
|---|---|---|
| Ethernet (10Mbit) | 1 MB/s | 1 ms |
| Myrinet | 125 MB/s | 10 µs |
| SCI | 500 MB/s | 1 µs |
| Spider (SGI) with 16 nodes | 12.8 GB/s | 156 ns |

- Programming a NOW
  - Like distributed systems: CORBA, COM+, RPC
  - PVM (Parallel Virtual Machine)
  - MPI (Message Passing Interface)



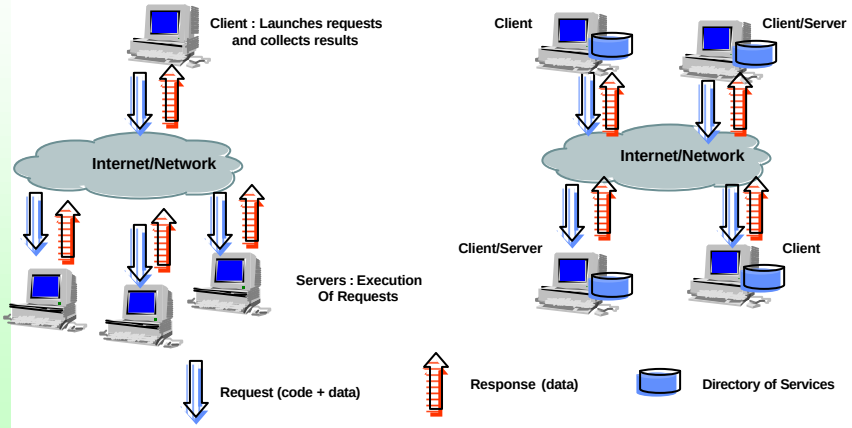


## Grid Computing(2)

- A Grid is likely to contain over 100 nodes
- A Grid is typically owned by a single organization or consortium (as a consequence, configurations are rather stable)
- Tool Kit: Globus (Global Grid Forum http://www.gridforum.org)
- According to their usage, several types of
- Grid can be distinguished:
  - Compute Grids (numerical intensive applications)
  - Data Grids (access and mangement of large data sets)
  - Instrumentation Grids (related to large and distributed experiments)
  - Application Grids (providing to users a set of applications)
- Refer to specialized work such as [FOS03]

# Global Computing (GC) and Peer-to-Peer (P2P)

- **Very large number of interconnected systems (not owned by a single organization and so little configuration stability)**
- **Global Computing and Peer-to-Peer Models (from [CAP02])**




a) - Global Computing

B) – Peer-to-Peer





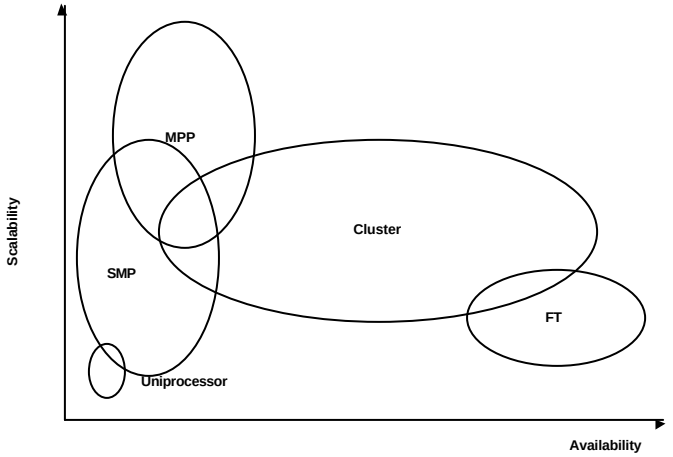
# SMPs, Clusters and MPPs - a Summary

- **Summary of the characteristics of SMP, Cluster and MPP Systems**

| Characteristics | SMP | Cluster | MPP |
|---|---|---|---|
| **Acceleration (speed up) or Increase (scale up)** | Scale Up | Scale Up | Speed Up |
| **Load-balancing** | Implicit | Requires software intervention | Requires software intervention |
| **High availability** | Typically not | Principal objective | Possible (is generally not an objective) |
| **Large configura tions (100 processors and beyond)** | Limited availabil ity in commodity technology; proprietary hardware is needed for large configurations | Limited by the characteristics of the interconnect network (often of commodity technology) | Principal objective (custom interconnect network) |
| **Single System Image** | Complete (by definition) | Limited | Limited |
| **Resource Sharing** | All (including the memory and the operating system) | Limited (typically discs and network connections) | Limited (typically just network connections) |
| **Programming** | Single process or multiple processes and threads allowing exploitation of parallelism | Custom programming necessary insofar as the objective is to exploit parallelism | Custom programming necessary in order to exploit parallelism (a more crucial issue for MPPs than for clusters) |
| **Flexibility in integrating different generation technologies** | Very limited | Yes | Limited |
| **Ease of maintenance** | Limited (often implies first stopping the system) | Easy (no need to stop the system) | Easy (no need to stop the system) |

## Positioning of the various options of architecture

- **Positioning of the various options of architecture (inspired by Compaq)**

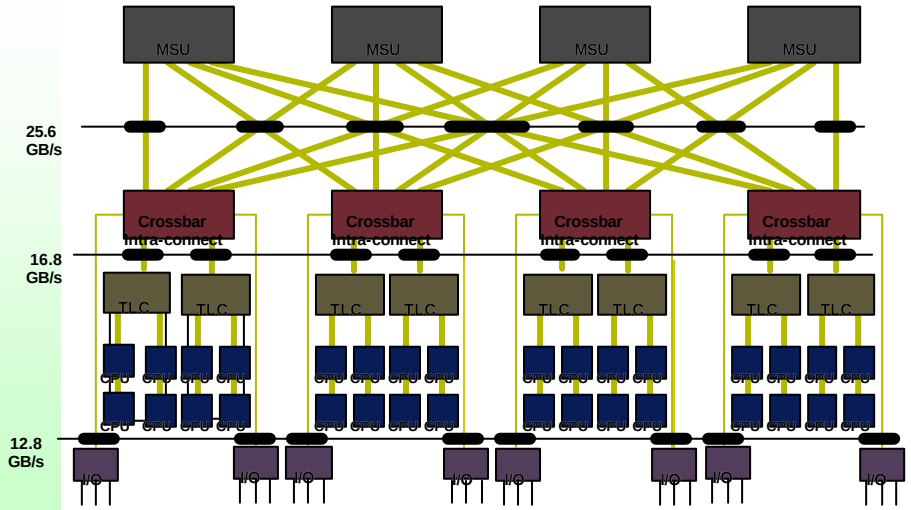



Page 35



## Flexible SMP-Cluster Architecture

- **Unisys' flexible ES7000 architecture (source Unisys)**




Page 36

## Flexible SMP-Cluster Architecture(2)

- **Bull NovaScale (Source Bull)**




## Vector Architecture Example: the NEC SX-6

- **NEC SX-6 Architecture**
  - **CMOS-based vector machine**
  - **Example of vector code:**

| Source code: | Machine code on a non-vector machine: |
|---|---|
| DO i=1, 256<br>Z(i)=X(i)+Y(i)<br>ENDDO | initialize I =1<br>test: see if the loop has finished; if so, branch to the end of the loop<br>calculate X(i) + Y(i)<br>place the result in Z(i)<br>increment i<br>jump to label test<br>end of loop |

A vector computer has vector registers intended to store vectors (indicated as r-vN) along with instructions which operates on vectors .
Vector machines have very sophisticated memory architecture to feed the vector register (256 GB/S bandwidth per node in the SX-6)
Example of machine code on a vectore machine:

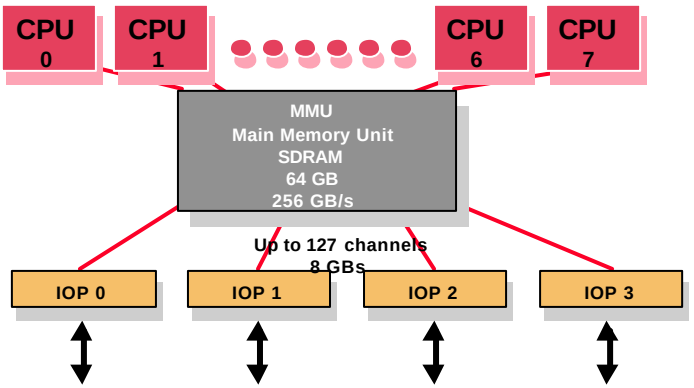
Load the 256 values of X[] into r-v1
Load the 256 values of Y[] into r-v2
Perform the vector addition of r-v1 and r-v2 into r-v3
Store the 256 values in r-v3 into Z[]

## Vector Architecture Example: the NEC SX-6(2)

- **SX-6 Family Characteristics**

| | Single node | Multiple nodes (cluster) |
|---|---|---|
| Peak performance | 64 Gflops | 8,192 Tflops |
| Number of processors | 8 (each of 8 Gflops) | 1024 |
| Number of nodes | 1 | 128 |
| Maximum memory size | 64 GoB | 8 TB |

- **Architecture of an SX-6 node (after NEC)**

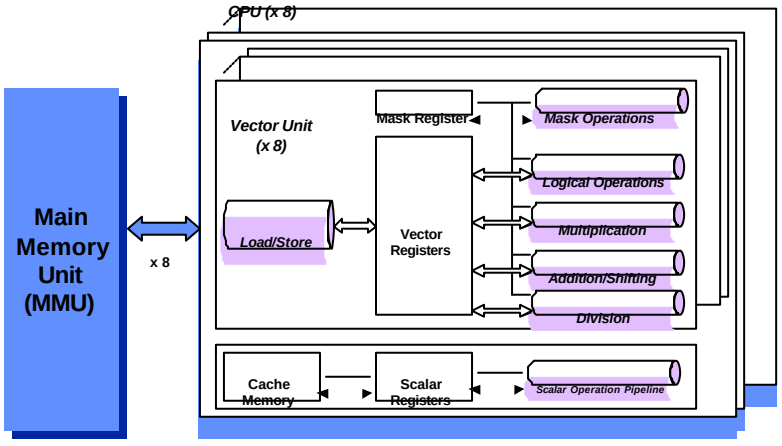






## Vector Architecture Example: the NEC SX-6(3)

- **Internal Architecture of an SX-6 processor (after NEC)**



# References


[CAP02] Franck Cappello et al. « Calcul Global et Pair à Pair »
http://www.lri.fr

[GAL97] Mike Galles, «Spider: A High Speed Network Interconnect»,
IEEE Micro, January/February 1997, pp. 34-39

[FOS03] Ian Foster (Ed.), Carl Kesselman (Ed.) "The Grid 2 – Blue Print for a New Computing Infrastructure"
2nd Edition, Morgan Kaufmann Publisher December 2003

[IBM03] IBM Report "IBM e Server pSeries SP Switch and SP Switch2 Performance"
February 2003, available on IBM Web site

[PAT04] David A. Patterson, John L. Hennessy, Computer Organization & Design: The Hardware/Software Interface
Morgan Kaufmann, San Mateo, Third Edition 2004

[SWE02] Mark Sweiger "Scalable Computer Architectures for Data Warehousing"
www.clickstreamconsulting.com