Server Architectures: Introduction

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 (this document)
  - Selection Criteria (review of needs)
  - Characterization of Server Usages (transaction processing, decis ion support and Web)
  - Introduction to Architecture Options
  - Dimensions of Growth
  - Parallelism
  - Historical Notes about System Architecture
- Processors and Memories
- Input/Output
- Evolution of Software Technologies
- Symmetric Multi-Processors
- Cluster and Massively Parallel Machines
- Data Storage
- System Performance and Estimation Techniques
- DBMS and Server Architectures
- High Availability Systems
- Selection Criteria and Total Cost of Possession
- Conclusion and Prospects

Selection Criteria

- Availability of Applications and Development Tools
- System Availability
- Data Integrity
- Security
- Performance
- Scalability (processing, storage, communication)
- Price (Total Cost of Ownership or TCO)
- Support for the Client/Server Model
- Architectural Maturity
- Investment Risk

Transaction Processing and Decision Support

A comparison of the characteristics of Transaction Processing (OLTP - On Line Transaction Processing) and Decision Support (DSS - Decison Support Systems)

Transaction Processing	*Decision Support*
❑ Sharing of information ❍ Both read and write, by the collection of users ❍ Must obey ACID properties	❑ Sharing of information ❍ Mostly read. ❍ Specialized database (different from database used in production) and speciallized databases (Datamarts)
❑ Irregular Workflow ❍ Pre-established set of functions typically O(100)	❑ Irregular Workflow ❍ No set of pre-established functions
❑ Simples Functions ❍ Not very complex - typically 10^5-10^7 instructions plus 10 I/ Os	❑ Complex Functions ❍ Frequently, complex requests making use of very large amounts of data
❑ Batch Processing ❑ Possible provided ACID properties are maintained	❑ Batch Processing ❑ Used for particularly long requests
❑ Very large number of users (several thousand, or several tens of thousands)	❑ Relatively small number of workstations
❑ Intelligent clients - workstations, other systems (servers, network terminals)	❑ Intelligent clients - workstations)

Transaction Processing and Decision Support(2)

Transaction Processing	*Decision Support*
❑ High Availability ❑ Typical requirement ❑ Recovery relies on ACID properties	❑ High availability ❑ Not normally a requirement ❑ Rather, the time to create or re- create the database is an important parameter
❑ Database size ❍ Proportional to the amount of business the company does	❑ Database size ❍ Proportional to the amount of time the company has been in business
❑ Very little data "touched" in any one transaction	❑ A great deal of data "touched" by any one request
❑ Automatic Load Balancing	❑ No Load Balancing
❑ Short response time and throughput guarantee (achieved through inter-request parallelism)	❑ Shorter response time is better (achieved through intra-request parallelism)
❑ Scalability is a typical requirement	❑ Scalability is a typical requirement

Characteristics of the Web

- Two main categories (non-exclusives) of Web server use:
  - Document servers providing research and navigation capabilities (as example, search engines)
  - Transaction processing server handling commerce orders (e-commerce: order, follow-up, invoice)
    Note : The categories are not exclusive because a user who wants to place an e-commerce order will likely first do a search for available bargains.
- Several studies have shown the characteristics of Web sites such as
  - Martin F Arlitt, Carrey L Williamson " Web Server Workload Characterization : The Search for Invariants " Department of Computer Science University of Saskatchewan March 1996
  - James E Pitkow « Summary of WWW Characterizations » Xerox Palo Alto Research Center 1998
  - Daniel A Menascé et al. « In Search of Invariants for E-Business Workloads » Proc. Second ACM Conference on Electronic Commerce, Minneapolis MN, October 17-20, 2000

Page 7

Characteristics of the Web(2)

- Invariants of e-commerce from Menascé/Almeida 2000
  - Based on the observation of 2 e-commerce Web sites:
    - Books and related items (shop based only on e-commerce)
    - Auctions concerning Internet domain names
- Results :
  - Most of the sessions last for less than 1 000 seconds
  - More than 70% of the functions executed are related to product selection
  - File accesses follows a Zipf distribution related to the file popularity: of the number of accesses. N, the number of accesses to file whose popularity rank is r is given by:
    - N = k / r (where k is a positive constant)
  - 16% of accesses are generated by Web robots
  - 88% of the sessions comprise fewer than 10 requests
  - The distribution of number of requests per session is “long-tailed” - that is, even if the average number is low, there is a non-zero probability of there being a large number of requests in a session

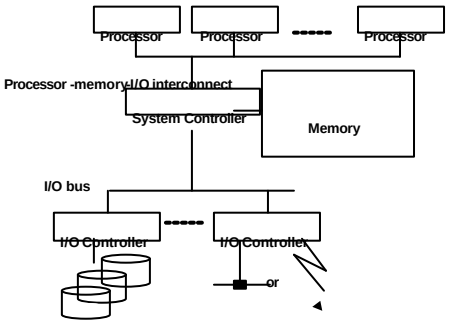
Page 8

Introduction to architectural options

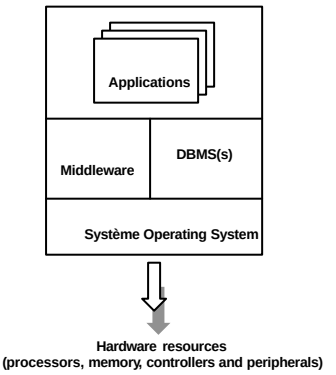
- **Search for scalability through multiprocessor architectures and parallelism**
- **Two families of multiprocessor architectures:**
  - **Tightly Coupled or Symmetric Multiprocessor (SMP) : all processors are sharing a shared and coherent memory as well as the whole set of system resources (memory, I/O devices). A single Operating System is controlling the system managing all system resources.**
  - **Loosely Coupled: a system is constructed by interconnecting (using a fast local area network technology) some number of independent systems, each having its own resources (processors, memory, I/O) and running under the control of its own copy of the operating system. Each such system is called a node. Clusters and Massively Parallel (MPP) systems both fall into the loosely-coupled classification. The term loose coupling refers to that fact there is no shared hardware in such systems (except for the interconnect and - in some systems - the storage subsystems).**

Introduction to architectural options(2)

- **SMP Architecture**



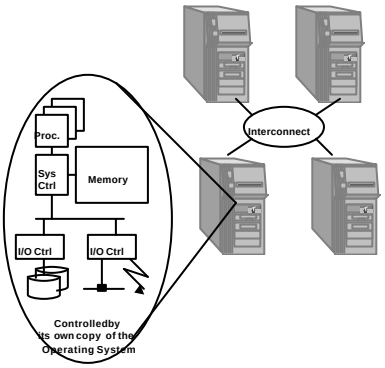
a) Hardware view of a Symmetrical Mutiprocessor



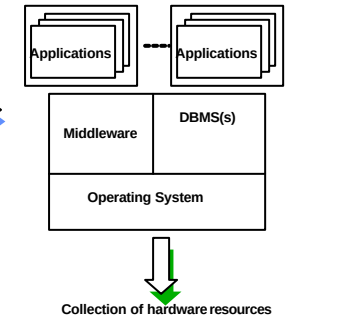
b) Software Viewof a Symmetrical Multiprocessor

Introduction to architectural options(3)

- **Loosely Coupled Architecture (Cluster/MPP)**



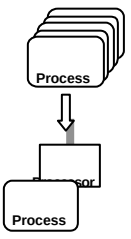
a) Loosely- coupled Architecture



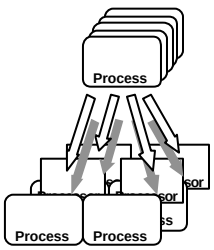
b) Idealized Viewof a Loosely-Coupled Architecture

Introduction to architectural options(4)

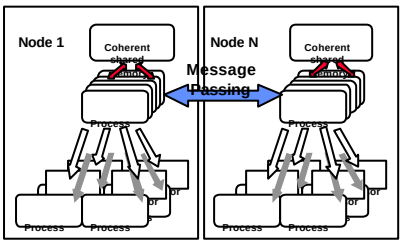
- **Execution and Programming Models**



a) - Uniprocessor



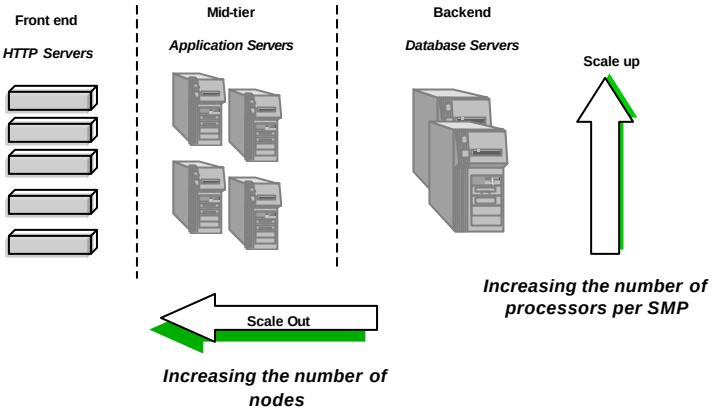
b) - Tightly-Coupled Multprocessor /SMP



c) - Loosely-Coupled Processor

Dimensions of Growth

- **Scaling Options (not exclusive)**



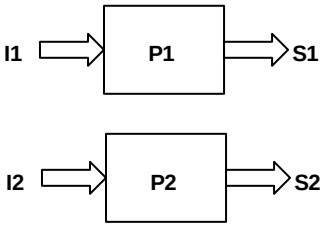
Page 13

Parallelism

Page 14

Parallelism – Definitions(1)

- **Philip Bernstein's Conditions (1966)**



  - Programs P1 and P2 may be executed in parallel (denoted as P1||P2) if, and only if, the following conditions are observed:
    - {I1∩S2 = ∅ , I2 ∩S1 = ∅ , S1 ∩S2 = ∅ }
  - More generally, a collection of programs P1, P2,... Pn is executable in parallel if, and only if, the Bernstein conditions are satisfied. In other words:
    - Pi||Pj " {i,j} with i≠j
  - Granularity of parallelism: coarse grain or fine grain

Parallelism – Definitions(2)

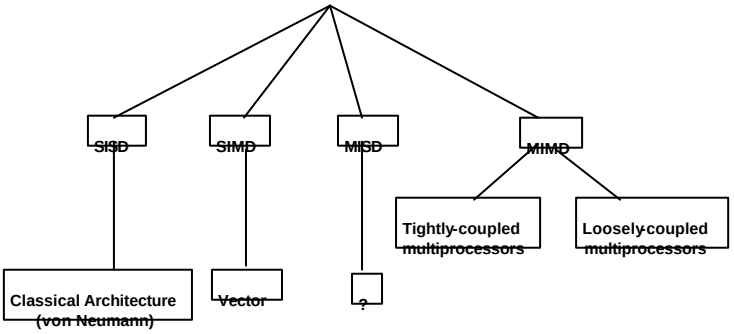
- **Sources of parallelism**
  - **Data Parallelism: the same operation is carried out - by different processors - on disjoint sets of data**
  - **Control Parallelism. different operations are carried out simultaneously. This sort of parallelism is available when the program is constructed of independent portions, or when certain control structures (such as loops) are likely to be carried out in parallel**
  - **Flow Parallelism work on one portion of the data flow is overlapped with work on another portion, i.e. a following operation can be started before the preceding one is finished. This is an assembly line or the pipeline model**

Parallelism – Definitions(3)

- **Flynn's Classification of Parallel Architectures (1972)**
  - **SISD (Single Instruction Single Data) in which just one stream of instructions performs a transformation on a single stream of data (John von Neumann)**
  - **SIMD (Single Instruction Multiple Data) in which the same stream of instructions is applied to disjoint sets of data**
  - **MISD (Multiple Instruction Single Data) in which several instruction streams are applied to the same stream of data**
  - **MIMD (Multiple Instruction Multiple Data) in which independent instruction streams are applied to independent sets of data. As it will be seen in the case of parallel DBMSs, it is the same program (the DBMS) which is simultaneously executed (but without synchronism) on disjoint sets of data, and one speaks then of SPMD (Single Program Multiple Data Stream)**

Parallelism – Definitions(4)

- **Relationship between Flynn's classifications and architectural options**



Limits of Parallelism

- **Amdahl's Law**
  - **Amdahl's Law expresses the speedup as a function of:**
    - **the fraction of the time that the computation spends in the part capable of improvement (1- a)**
    - **speedup of the improved part (P being the improvement factor)**

$$Maximum\_Speedup \quad \leq \frac{1}{a + \frac{(1-a)}{P}} \leq \frac{1}{a}$$

- **If the sequential portion of the application is 10%, and we assume we have ten processors in parallel, the maximum speedup is 5.26x, although with an infinite number of processors a 10x speedup would be achievable..**
- **Available speedup falls quickly as the sequential portion of the application grows. As an example, if the sequential portion were to be 20%, then (still with 10 proces sors) the available speedup is 3.57x falling to 2.17x when the sequential portion grows to 40% and to 1.67x at 60% sequential.**

Limits of Parallelism(2)

- **Gustavson's Law:**
  - **In some intensive numerical calculations the sequential portion may be reduced almost to zero by increasing the size of the problem enough:**
  - **Let a program with a sequential portion s and whose execution time a for the portion likely to be parallelizable is a linear function of the size of the problem n.**
    - **The execution time on a single processor would then be:** $t = s + a \times n$
    - **Amdhal's Law gives the execution time on a system with n processors:**

$$t = s + \frac{a \times n}{n} = s + a$$

    - **And speedup (which tends to n as a tends to infinity) is:**

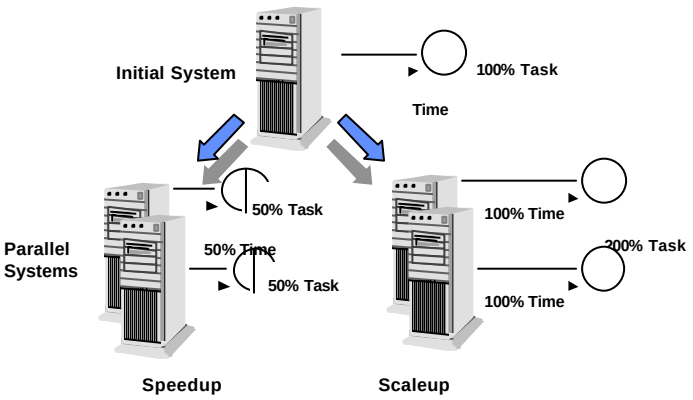
$$A = \frac{s + a \times n}{s + a}$$

Speedup and Scaleup

- **An ideal parallel system must have two properties: linear speedup and linear scaleup [DEW92]**
  - **a linear speedup if n times more resources make it possible to treat a given task in n times less time than the reference system (Decision supports Systems generally seek for speedup through intra-request parallelism)**
  - **a linear scaleup if n times more resources make it possible to deal with an n times larger problem in the same time as the reference system (Scaleup performance increases for OLTP systems are done through inter-request parallelism)**

Speedup and Scaleup(2)

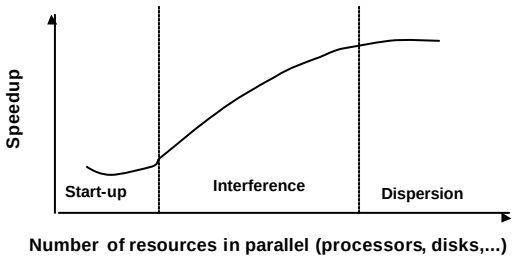
- **Illustration (ideal case)**



Note: For most of the OLTP applications, linear Speedup and linear Scaleup can't be achieved due to the necessary synchronization for data accesses

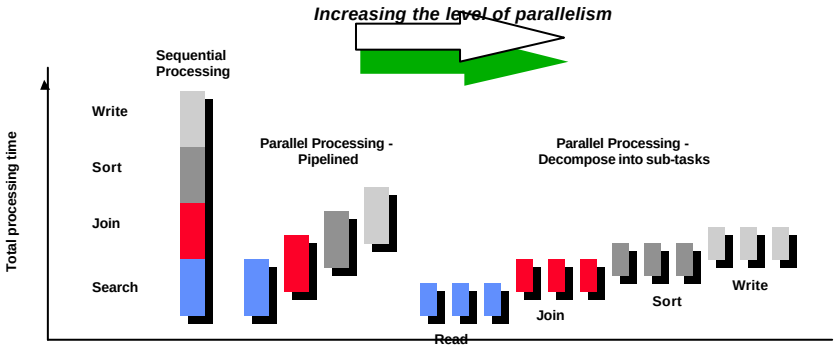
Speedup and Scaleup(3)

- **The behavior of real systems diverges from the ideal case of linear acceleration**



Parallelism and DBMSes

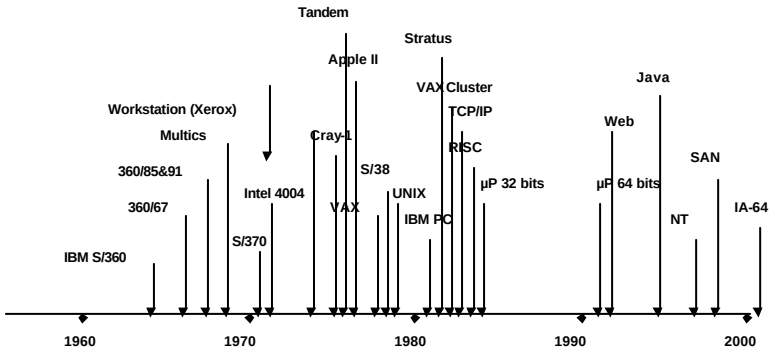
- **Parallelization of an SQL Request (according to Informix)**



Note: This subject is developed into the « DBMS and Server Architectures » presentation

Historical Notes about System Architecture

- **Key System Architecture Milestones**



Page 25