CS3 Database Systems Note 16

Introduction to Parallel Databases

Thomas Zurek

December 5, 1995

- Motivation
- Some Workloads
- Example of a Workload: Joins
- Parallel Hardware
- Example of Parallelising a Workload: Parallel Joins
- Other Types of Parallelism
- Oracle 7

Why Parallel Databases?

- bottlenecks (I/O, CPU)
- performance gap: CPU – I/O
- cheaper (?)

Jim Gray:

- application pull
- technology push

Typical Workloads

- lots of transactions
- complex transactions
- lots of queries
- complex queries
- lots of operations

▷ 'huge' operations

Reminder

DBS_Student		
Name	**Mark 1**	**Mark 2**
Peter	B	A
Garry	A	B
Mary	A	A

Address		
Name	**Street**	**Phone**
Mike	High Street	123 4567
Peter	Mayfield Road	222 3333
Conny	George Street	987 8888
Garry	Teviot Place	999 4321
Mary	Melville Drive	555 6666

Reminder

in SQL

```
SELECT *
FROM   DBS_Student R, Address S
WHERE  R.name = S.name
```

in Relational Algebra

$$R \bowtie_C S \quad \text{with} \quad C \equiv (R.name = S.name)$$

Nested-Loop Joins

for each r in R do
    for each s in S do
        if (C) then
            put $r \circ s$ into result

$|R| + |R| \cdot |S|$ tuple accesses

$$\begin{aligned} |R| &= 100,000 \\ |S| &= 100,000 \\ \mu &= 25 \text{ ms} \end{aligned}$$

Algorithm	**I/O Accesses**	**Example**
Nested-Loop	$\lvert R\rvert + \lvert R\rvert \cdot \lvert S\rvert$	$\approx$ 8 years
Sort-Merge	$\lvert R\rvert + \lvert S\rvert$	$\approx$ 83 min
Index-Join	$\lvert R\rvert$ to $\lvert R\rvert + \lvert S\rvert$	$\approx 42 - 83$ min

$\longrightarrow$ **I/O bottleneck**

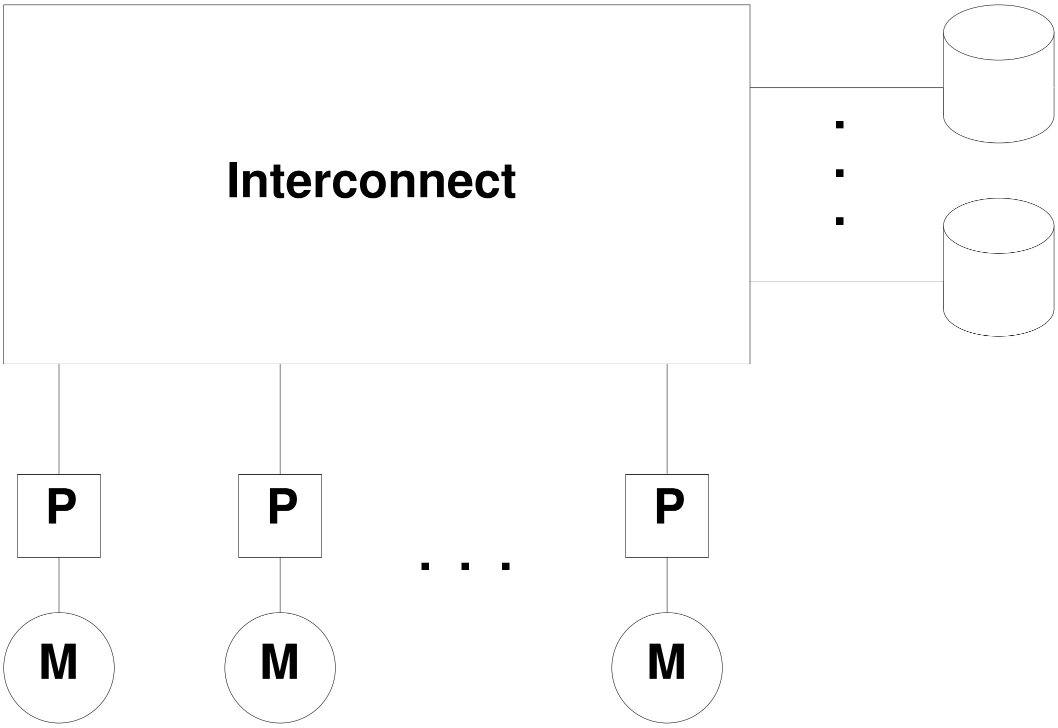
The fundamental idea:

Magnify the raw power of individual commodity components by integrating these in a complete system along with the appropriate parallel database software.

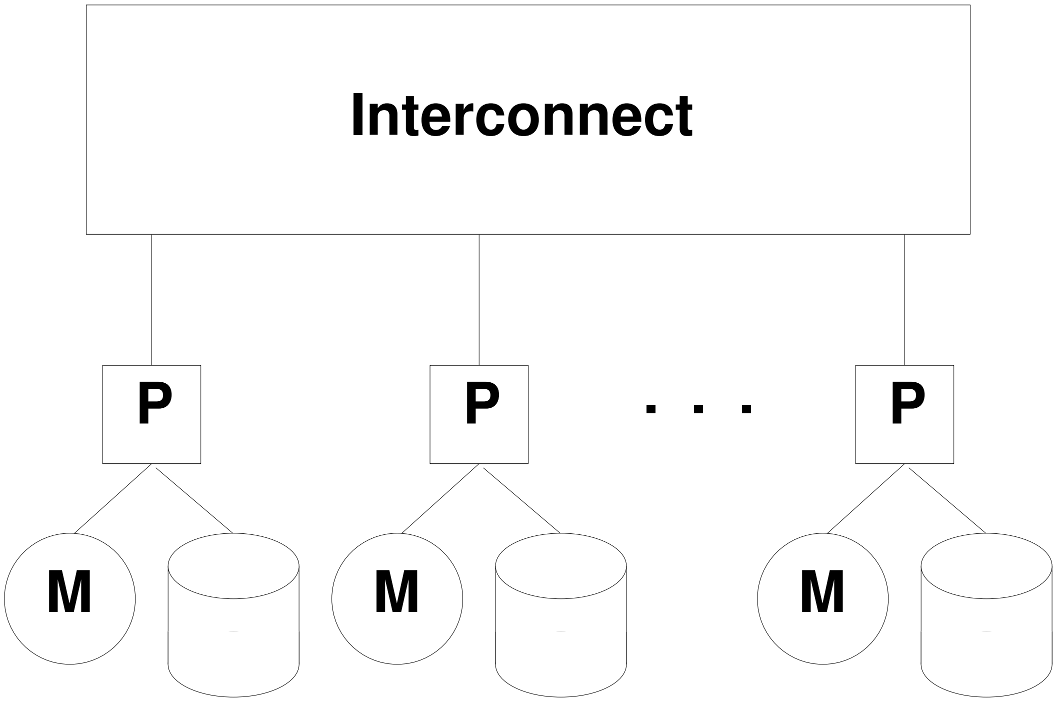
Expected Advantages

- scalability
- cheaper price/performance ratio

Shared-Disk Architecture



Shared-Nothing Architecture



$$R \bowtie S$$

$$\downarrow$$

$$R_1 \bowtie S_1 \ \cup \ \cdots \ \cup \ R_n \bowtie S_n$$

Data partitioning allows

- I/O parallelism
- CPU parallelism

Example

All tuples of R and S whose names start with

- A to node 1
- B to node 2
- ...

BUT:

more names start with an M than with a Q

$\downarrow$

load imbalance through data skew

- transaction level
  - inter-transaction parallelism
  - intra-transaction parallelism
- query level
  - inter-query parallelism
  - intra-query parallelism
- operator level
  - inter-operator parallelism
  - intra-operator parallelism

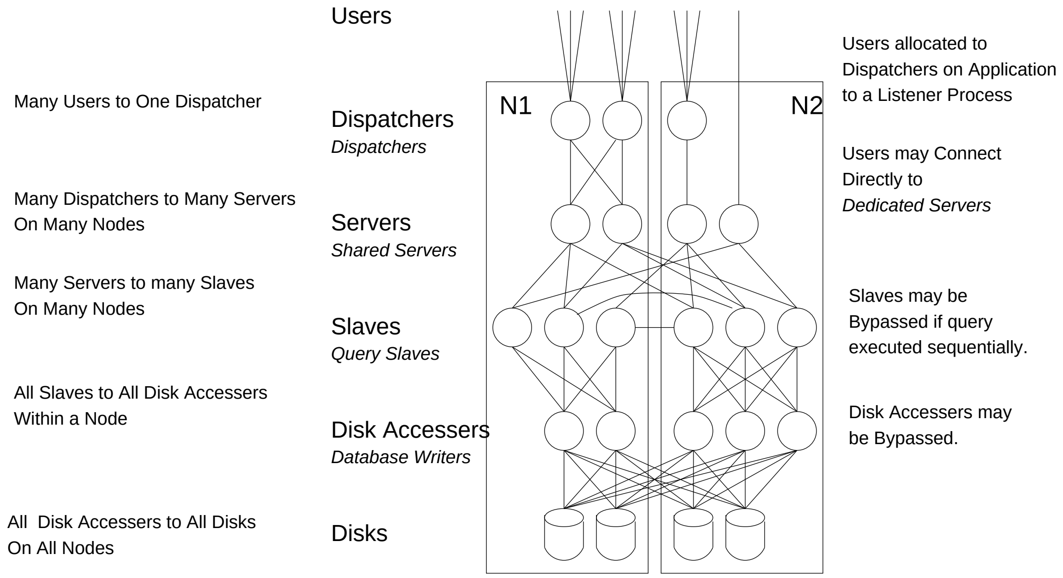
Software Architecture

- shared-disk notion
- targeted for many different platforms
- extremely flexible
- good load balancing properties:

  vertical + horizontal

  data movement is easy

Oracle 7


Users
Many Users to One Dispatcher
Dispatchers
Dispatchers
N1
N2
Users allocated to
Dispatchers on Application
to a Listener Process
Users may Connect
Directly to
Dedicated Servers
Many Dispatchers to Many Servers
On Many Nodes
Servers
Shared Servers
Many Servers to many Slaves
On Many Nodes
Slaves
Query Slaves
Slaves may be
Bypassed if query
executed sequentially.
All Slaves to All Disk Accessers
Within a Node
Disk Accessers
Database Writers
Disk Accessers may
be Bypassed.
All Disk Accessers to All Disks
On All Nodes
Disks