FROM PARALLEL DATABASES TO DISTRIBBUTED DATA MANAGEMENT IN HIGH PERFORMANCE NETWORKS

Lionel Brunie

Laboratoire de l'Informatique du Parallélisme
Ecole Normale Supérieure de Lyon
France

Lionel. Brunie@ens-lyon.fr
www.ens-lyon.fr/LIP/membres/lbrunie.us.html

OUTLINE

I- Parallel Databases : motivations

II- Parallel Databases : basics

III- A short overview of commercial parallel DBMS

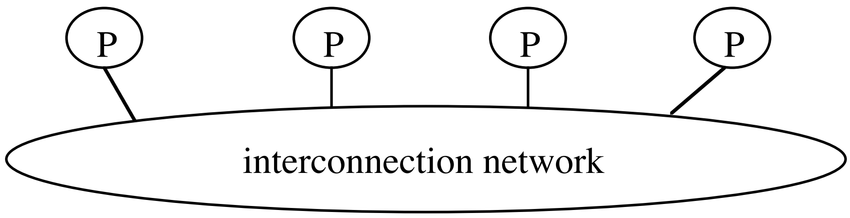
IV- When Parallel Databases meet High Performance Networks : opening application perspectives

V- Conclusion

WHAT IS A PARALLEL MACHINE ?

---> As many definitions as machines !

---> Two basic points
- **+ multi-processor**
- **+ interconnection network**



---> Objective : high performance

PARALLEL DATABASES : MOTIVATIONS

4 main motivations

---> necessity

---> price

---> new applications

---> theoretical interest

PARALLEL DATABASES : MOTIVATIONS I

A first motivation : necessity

---> IS size ↑↑

---> Transactional load ↑↑

===> memory and IO bottlenecks are getting worse !

---> Query complexity ↑↑

EXAMPLES OF TARGET APPLICATIONS : DECISION SUPPORT SYSTEMS

---> Relational databases

---> (Very) complex queries : -> 30+ joins

---> Very high computation time : -> hours

=> high memory and CPU requirements

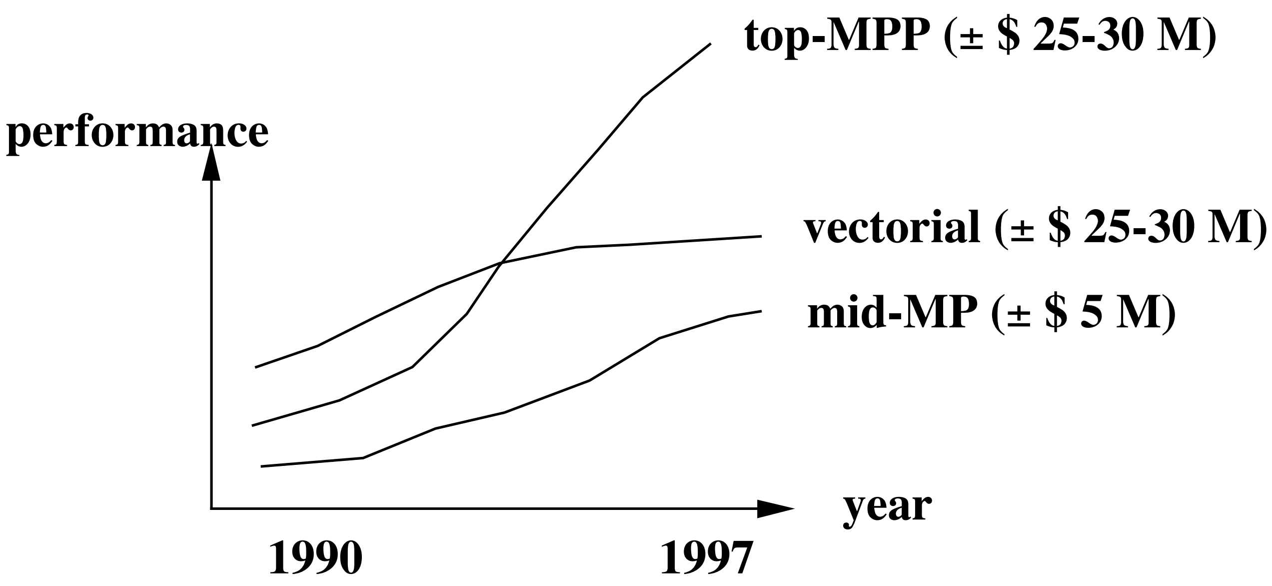
=> scalable architectures

PARALLEL DATABASES : MOTIVATIONS II

A second motivation : $$$

---> one order of magnitude

---> evolution :



PARALLEL DATABASES : MOTIVATIONS III

A third motivation : applications

---> data mining

---> decision support (data warehouse)

---> management of complex data

EXAMPLES OF TARGET APPLICATIONS : VIDEO SERVERS

---> Huge volume of raw data :
+ bandwidth : ± 150 Mb/s ; 1h ≈ 70 GB
+ bandwidth TVHD : 863 Mb/s ; 1h = 362 GB

+1h MPEG-2 video ≈ 2 GB

=> parallel I/O!!!

---> highly structured, complex and voluminous metadata (descriptors) : -> GB

=> large memory !!!

EXAMPLES OF TARGET APPLICATIONS : MEDICAL IMAGE DATABASES

**---> PACS : Picture Archiving and
Communication System**

**---> RIS : Radiological Information System
+ 1 year production ≈ 1 à 8 TB**

**---> Highly heterogeneous data (multiple
□□□□□□□□□□□□□□imaging devices)**

---> Heterogeneous queries

**---> Complex manipulations (e.g. multimodal
image fusion)**

=> high CPU, memory, IO demands

EXAMPLES OF TARGET APPLICATIONS : DOCUMENTARY DATABASES

---> Large libraries : -> 1,000,000 references

---> numerous indexes

---> performance requires in-memory work

=> high memory requirements

PARALLEL DATABASES : MOTIVATIONS IV

A fourth motivation : theoretical interest

---> query optimization

---> execution strategies

---> load balancing

---> applications

MOTIVATIONS : CONCLUSION I

Parallel architectures offer :

☐

+ CPU power : -> Tflops

+ parallel IO :
e.g. Oracle on nCube 512 : 528 MB/s
=> 1 TB scan : 31 mns

+ extensive memory :
128 (128 MB) nodes => 16 GB

+ availability

+ scalability

---> *With a very good perf:(hardware…) cost ratio!*

MOTIVATIONS : CONCLUSION II

---> However, parallelism still remains dedicated to (very) high performance applications

---> new challenge : everyday life parallelism

---> opportunity : high performance networks

PARALLEL DATABASES : BASICS

---> P□erformance criterions

---> Architecture paradigms

---> Implementations

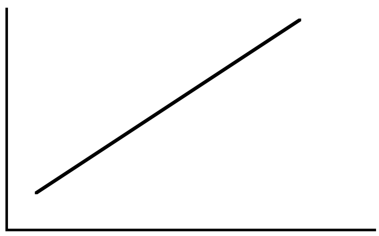
P□ERFORMANCE EVALUATION : SCALABILITY

+ *linear speed-up* : at constant DB size

configuration augmentation

=> linear augmentation of the performances

speed-up
=
new perf / old perf

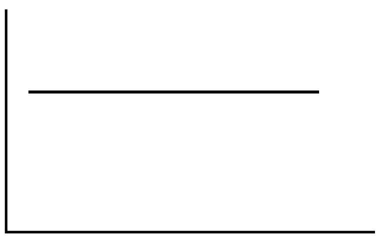


configuration size

+ *linear scale-up* : equal augmentation of the DB size and the configuration size

=> same performances

new perf / old perf



configuration size

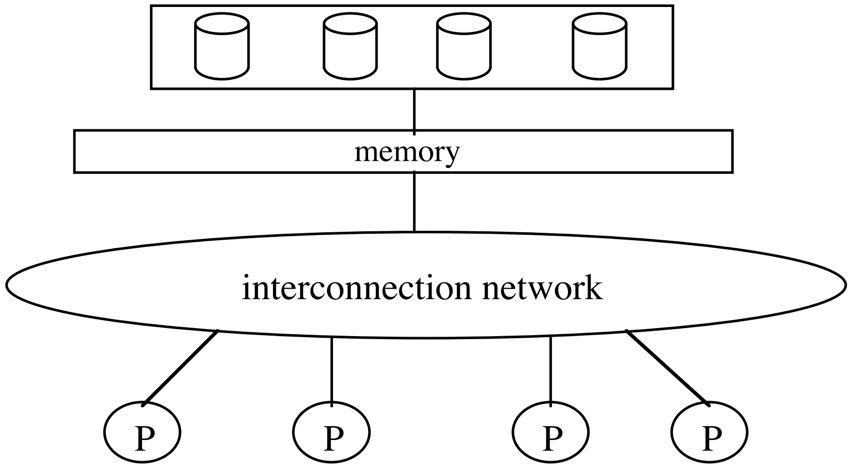
3 INTRINSIC LIMITATIONS

---> **Start-up time**

---> **Interferences (contentions)**
- **+ concurrent accesses to shared ressources**
- **+ sources of contentions :**
  - **- architecture**
  - **- data partitioning**
  - **- communication management**
  - **- execution plan**

---> **Load imbalance**
- **+ response time = slowest process**
- **+ balance data, IO, computations, communications**

SHARED MEMORY MACHINES



Advantages : "easy parallelism"

+ one single address space (memory - disks)
=> data transparently accessible
=> easier load balancing

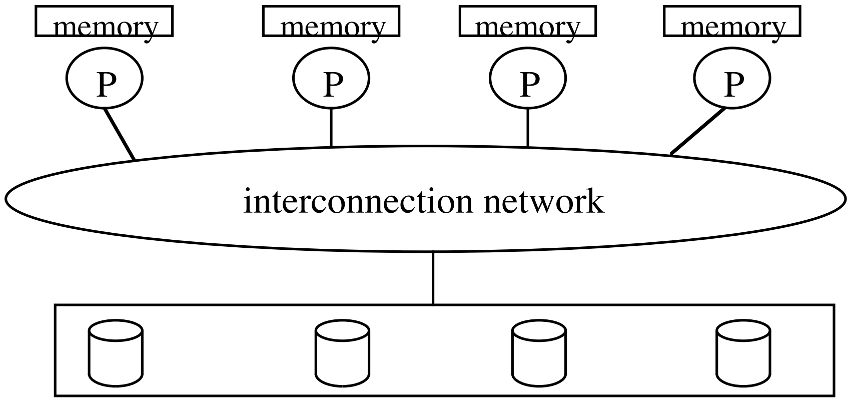
+ fast access to data (no communications)

Limitations : scalability/availability

+ memory and IO contentions

Exemples : XPRS (UCB), DBS3 (BULL), Sequent

SHARED DISKS MACHINES



Advantages :

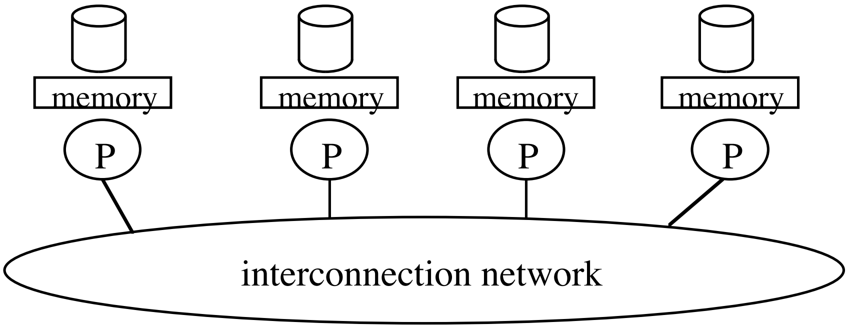
+ no memory contentions
+ easy migration from uni-processor server

Limitations :

+ cache consistence management
+ IO bottleneck

Exemples : DEC-RDB, IBM-DB2, Oracle

SHARED NOTHING MACHINES



Advantages :

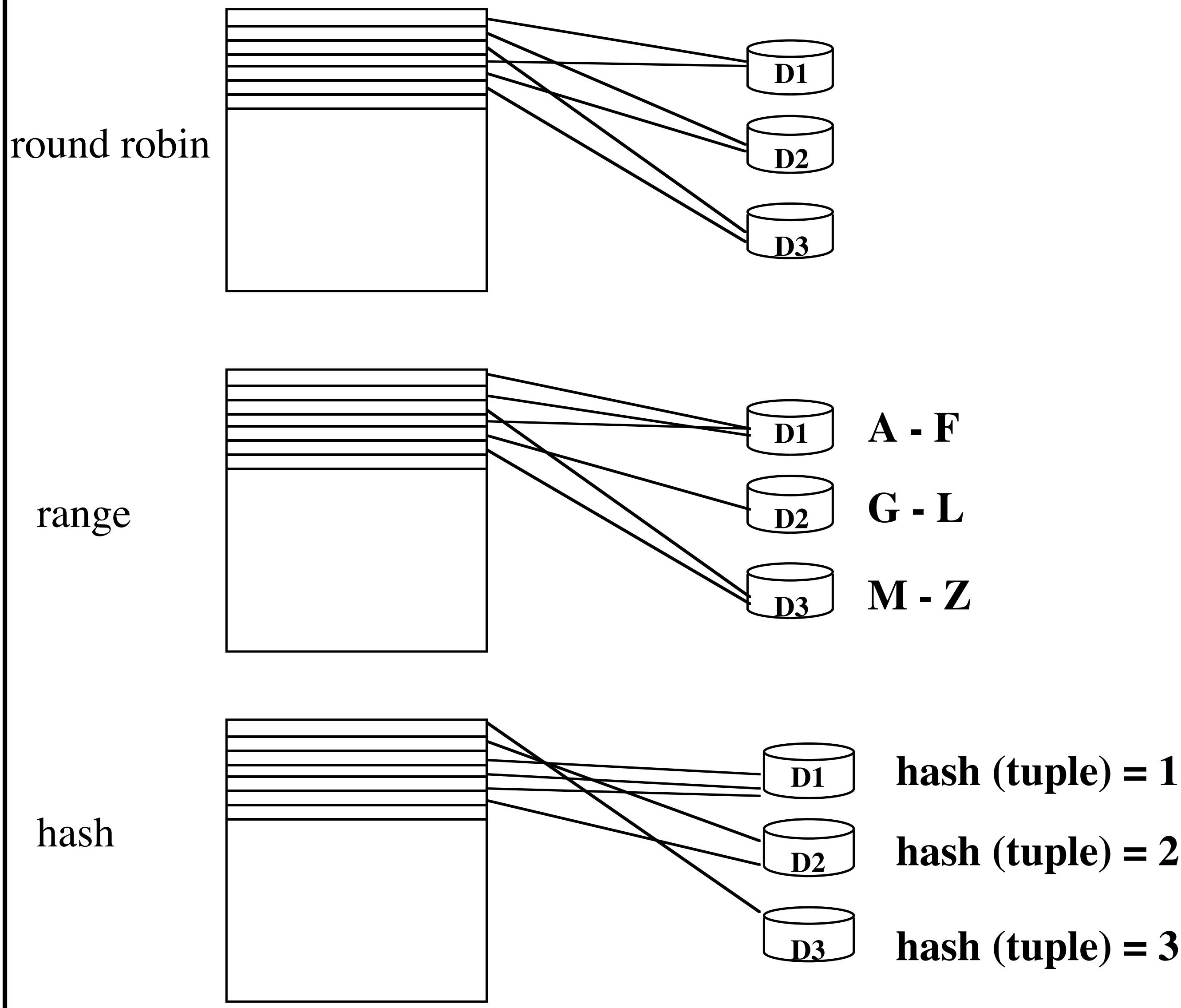
+ cost
+ scalability
+ availability

Limitations : parallelism management

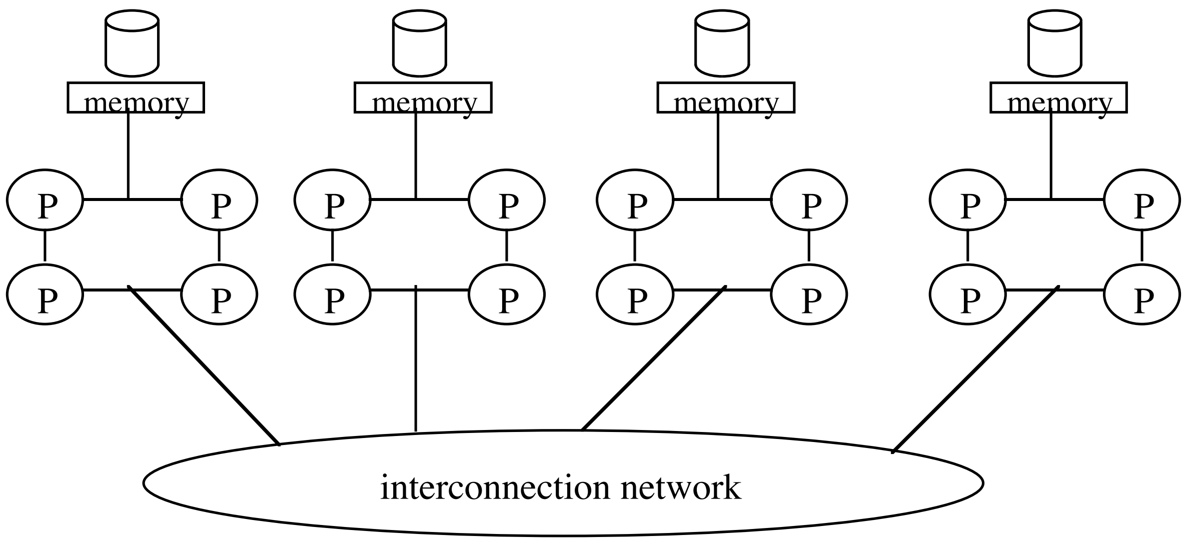
+ data partitioning
+ communication management
+ load balancing
+ complexity of the optimization process

Exemples : Gamma, Bubba, nCube, Teradata, Tandem

DATA FRAGMENTATION



CLUSTERS



Advantages : mixed approach

Limitations : complexity

Alternative : SN/SD clusters

Exemple : SGI, Matra Capitan, LAN subnets

PARALLEL APPLICATION

---> target architecture (virtual machine)

---> data model

---> (parallel) execution scenario

---> task scheduling

---> data mapping

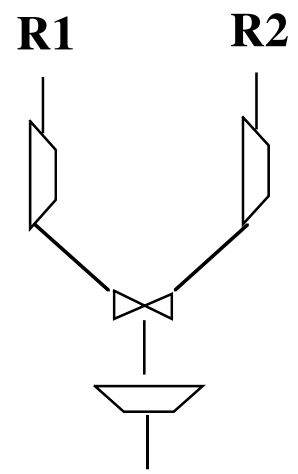
---> task mapping

---> execution engine

---> load balancing

---> monitoring/optimization

SOURCES OF PARALLELISM



---> pipeline parallelism

---> intra-operator parallelism

---> inter-operator parallelism

---> inter-query parallelism

PARALLEL EXECUTION SCENARIO

---> Operator processing ordering

---> degree of inter-operation parallelism

---> access method (e.g. indexed access)

---> characteristics of intermediate relations (e.g. cardinality)

---> degree of intra-operation parallelism

---> algorithm (e.g. hybrid hash join)

---> redistribution procedures

---> synchronizations

---> scheduling

---> mapping

---> control strategies

BASIC CONCEPTS : CONCLUSION

---> Shared nothing : high performance

---> Shared disk : easy migration

---> <u>*Key concepts*</u> : scalability,
load balancing,
parallel execution scenario

PARALLEL DBMS
VS
DISTRIBUTED DBMS

---> Basically
+ a D-DBMS = n (DBMS + com. layer)
+ a P-DBMS = 1 (DBMS ported on a parallel platform

---> A single query server : "one DB machine"

---> One parallel platform vs n uni-processors

---> High performances

---> ≠ algorithms

=> objectives/constraints are different

COMMERCIAL PARALLEL DBMS

---> Parallel DBMS :

+ Teradata (NCR)
- 1983
- originally : specific hardware
  + specific OS
- shared nothing
- decision support
- up to 400 nodes - 11 To
- 20 nodes - TPC-D/100GB :
  1442 queries per hour

+ Tandem Non-Stop SQL
- shared nothing
- transactional systems
- up to 64x4 nodes
- fault tolerance
- 112 nodes - TPC-C :
  22050 queries per second

COMMERCIAL PARALLEL DBMS
(cont'd)

---> Parallel Versions of Sequential DBMS :

SD	{	**+ Oracle**	**-> distributed locking**
SN	{	**+ Informix**	**-> migration**
		+ Sybase	**-> administration**
		+ DB2	**-> optimizer**

---> General tendencies

□□+ use of standard components
□□ + distributed memory architectures (scalability)
□□ +□interfaces with concurrent systems and various platforms
□□ + administration tools
+ multi-purpose (decision support, transactional, multimedia, video…)
+ fault tolerance
□□ + intra- and inter-parallelism
□ + load balancing strategies

HIGH PERFORMANCE NETWORKS I

---> High bandwidth : -> Gb/s

---> Low latency : -> µs

---> … Low price : -> $100-1000 per node

Sources of the miracle :

- **+ point-to-point**
- **+ switch based technologies**
- **+ custom VLSI (e.g. Myrinet)**
- **+ network protocols (e.g. ATM)**
- **+ kernel bypassing (e.g. VIA)**
- **+ network technology (e.g. optical fibre)**

AN EXAMPLE OF HPN : ATM

---> Targets :

- **+ multimedia**
- **+ isochronous data (video, audio)**
- **+ mixed traffic**
- **+ standard & interoperablity (ATM Forum)**
- **+ WAN/MAN/LAN**
- **+ bandwidth on demand**

---> Technology :

- **+ small constant-sized cells (53 bytes)**
- **+ cell relay**
- **+ hardware switching**
- **+ virtual channels**
- **+ statistical multiplexing**
- **+ quality of service**
- **+ up to 2488 Gbps (OC-48)**

AN EXAMPLE OF HPN : MYRINET

---> Targets :

- **+ parallel & distributed computing**
- **+ low latency**
- **+ data servers**
- **+ interoperability**
- **+ open software (source code)**

---> Technology :

- **+ switched network (crossbar)**
- **+ concurrent communications**
- **+ custom VLSI CMOS components**
- **+ up to 500 Mbps UNIX-UNIX**
- **+ theoretical : 1.28 + 1.28 Gbps**
- **+ up to 1 Gbps - 5μs BIP : user-level, zero-copy protocol, DMA**
- **+ $1600-1800 per host**

HIGH PERFORMANCE NETWORKS II

---> An opportunity for parallel databases :

+ WAN : connecting people to archives

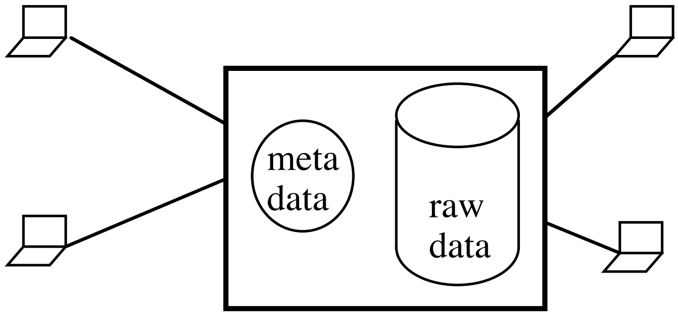
+ LAN : use your local network as a parallel machine… NOW

+ SAN : do-it-yourself and discount machines… PoPC

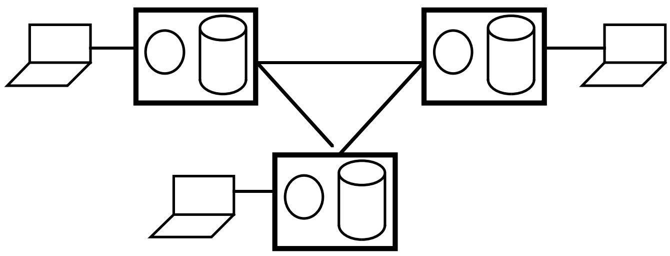
HIGH BANDWIDTH WAN

---> "bulk" applications :
e.g. video on demand

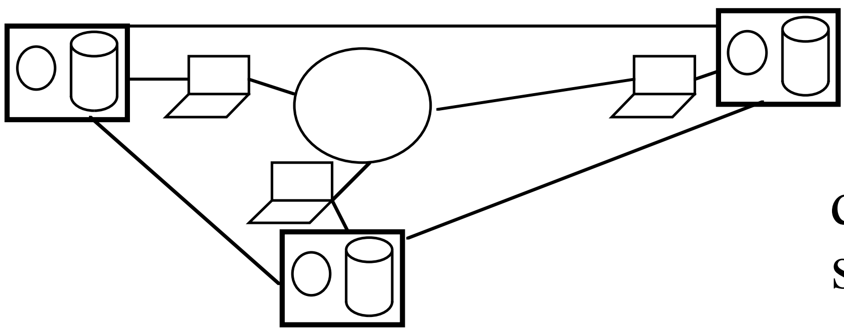
---> decoupling raw data and meta data



parallel
server



distributed
server



decoupled
server

HIGH PERFORMANCE LAN

---> Every lab owns a virtual parallel machine !

---> PC --> WS --> mainframes :

+ WS price:perf ratio : + 80% per year
+ supercomputers : +25% per year

---> convergence LAN - HPN

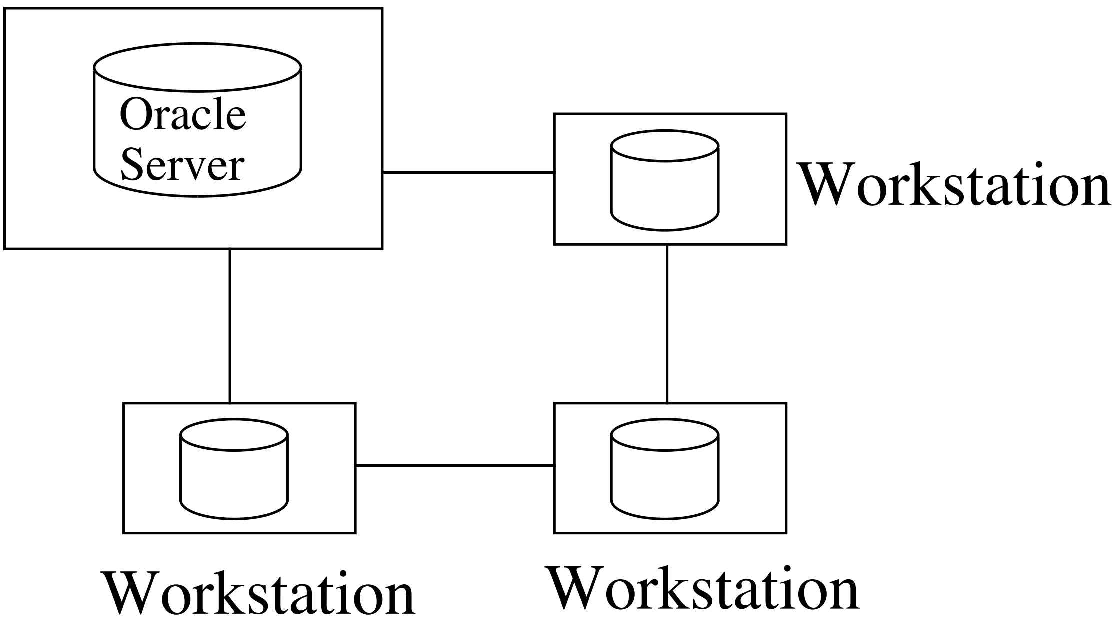
=> NOW ca be used :

☐

- as virtual super-servers
- as virtual parallel machines

USING HP LAN AS VIRTUAL SUPER SERVERS

---> Ex : Oracle // Server for poor labs :



---> read-only databases
---> "hot" sub-base

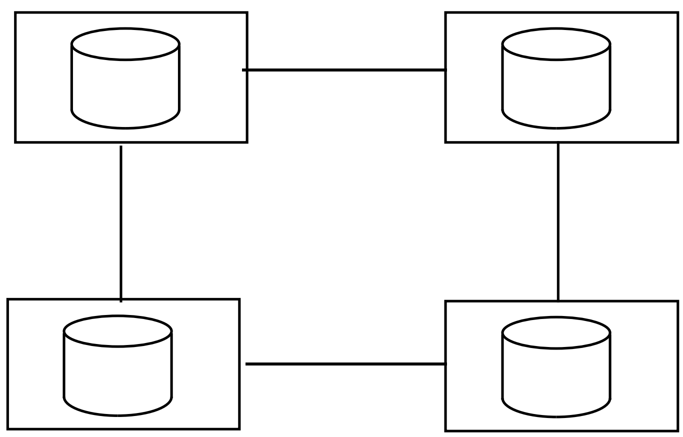
---> no perturbation (most WS are idle !)

USING HP LAN AS VIRTUAL PARALLEL MACHINES

---> Great opportunity for researchers :
SN and hierarchical machines
for (almost) free

---> opportunity for big sites :
ex : hospital PACS

cytology MRI



radiology nuclear imaging

HIGH PERFORMANCE SAN

---> **High bandwidth, low latency, SAN only *but* cheap (e.g. Myrinet)**

---> **first opportunity : do-it-yourself e.g. Beowulf project (NASA) :**

interconnecting 16 PCs :

1 GOPS
0.5 GB DRAM
20 GB disks
100 Mb Fast Ethernet

$50,000 !

---> **second opportunity : cheap Piles of PCs (PoPC)**

POPC : THE NEXT GENERATION OF // MACHINES

---> Standard components

---> High performance SAN

---> Little platforms (e.g. 8 processors)

---> Best price:performance ratio

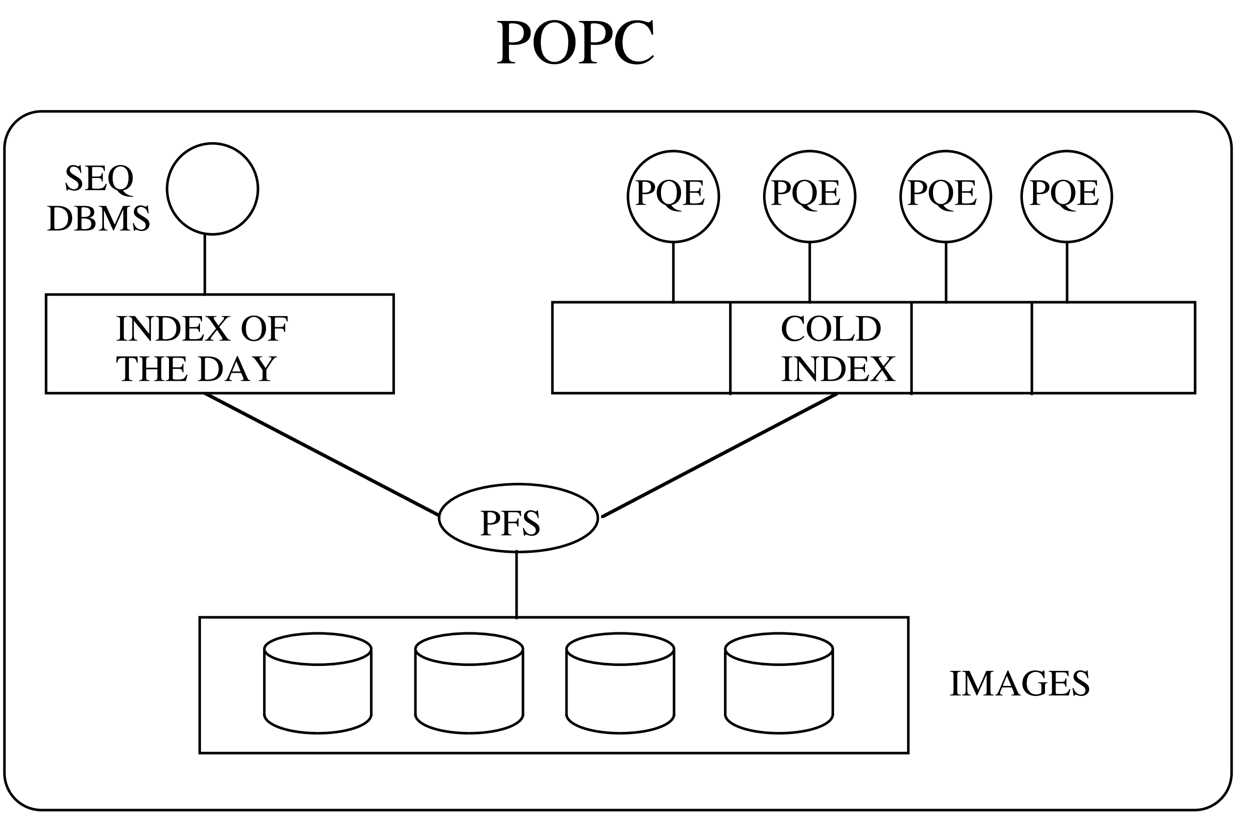
PoPC AND PARALLEL DATABASES

---> Budget restrictions !

---> Parallel DBMS (very !) expensive !

---> Parallel database applications
… without parallel DBMS

PARALLEL RIS ON TOP OF A POPC



PQE : Parallel Query Optimizer

CONCLUSION

---> New applicatons and new work loads demand high performances

---> High Performance Networks
=
opportunity
(-> cheap, scalable)

WHAT APPLICATIONS ?