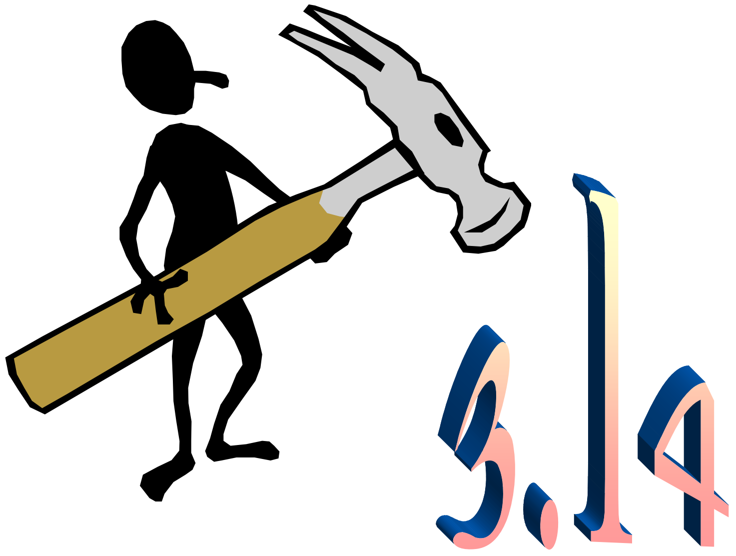
Parallel and Grid Computing

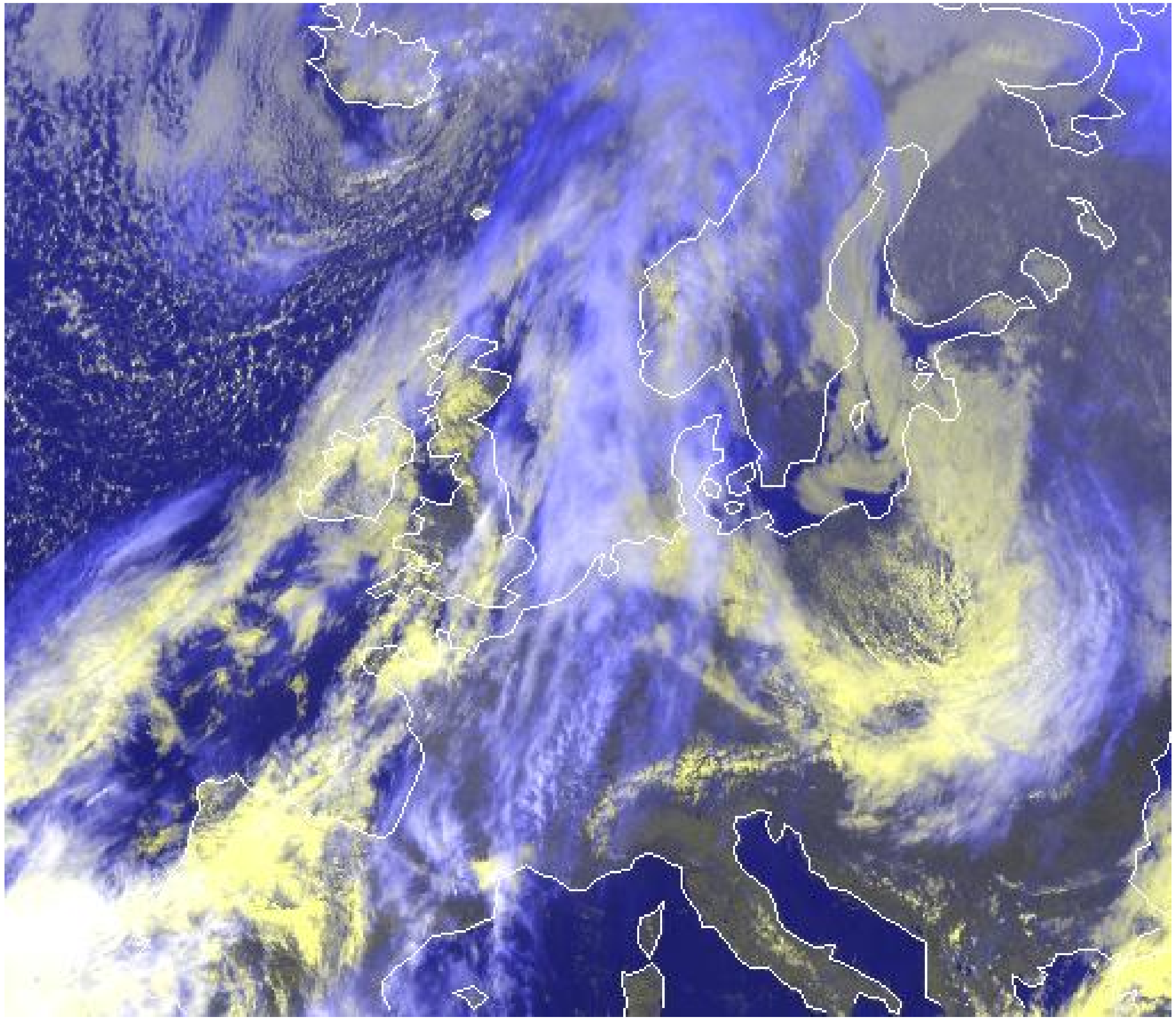
Brian Nielsen

bnielsen@cs.aau.dk

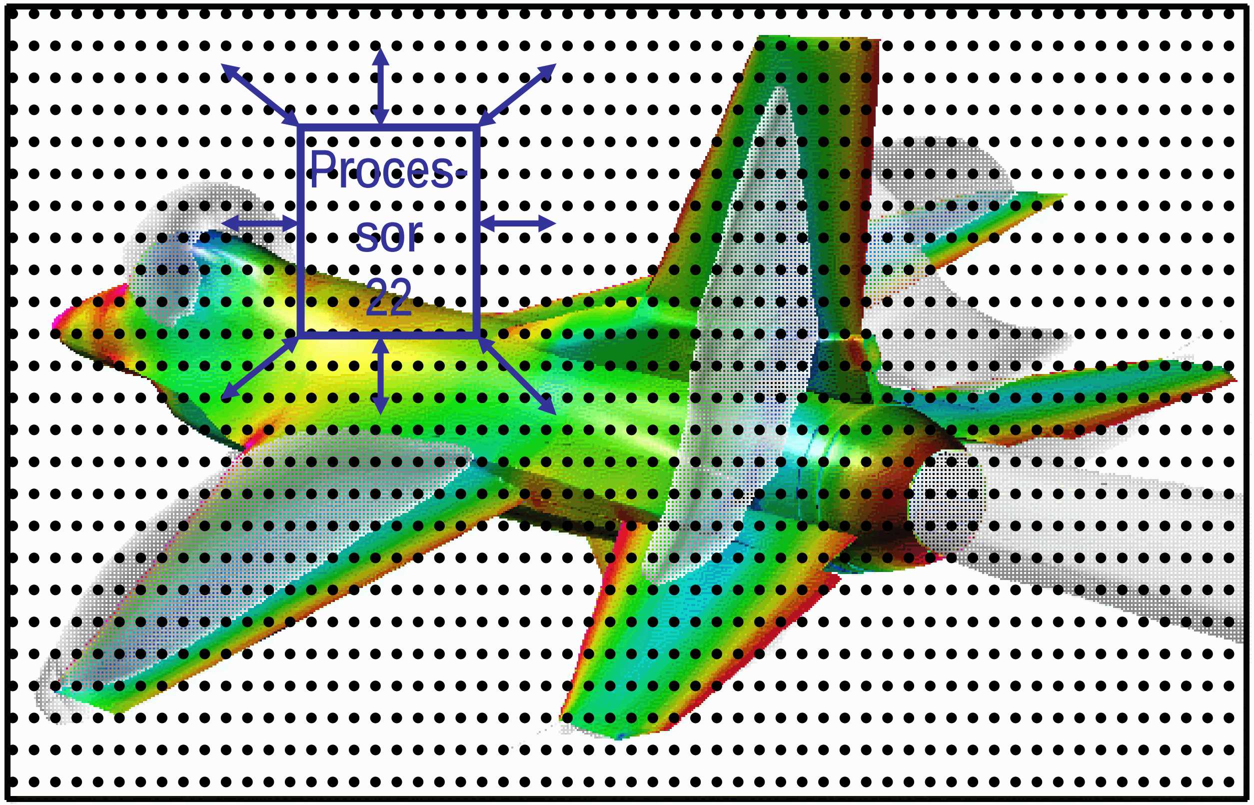
Number Crunching



Weather-forecasting

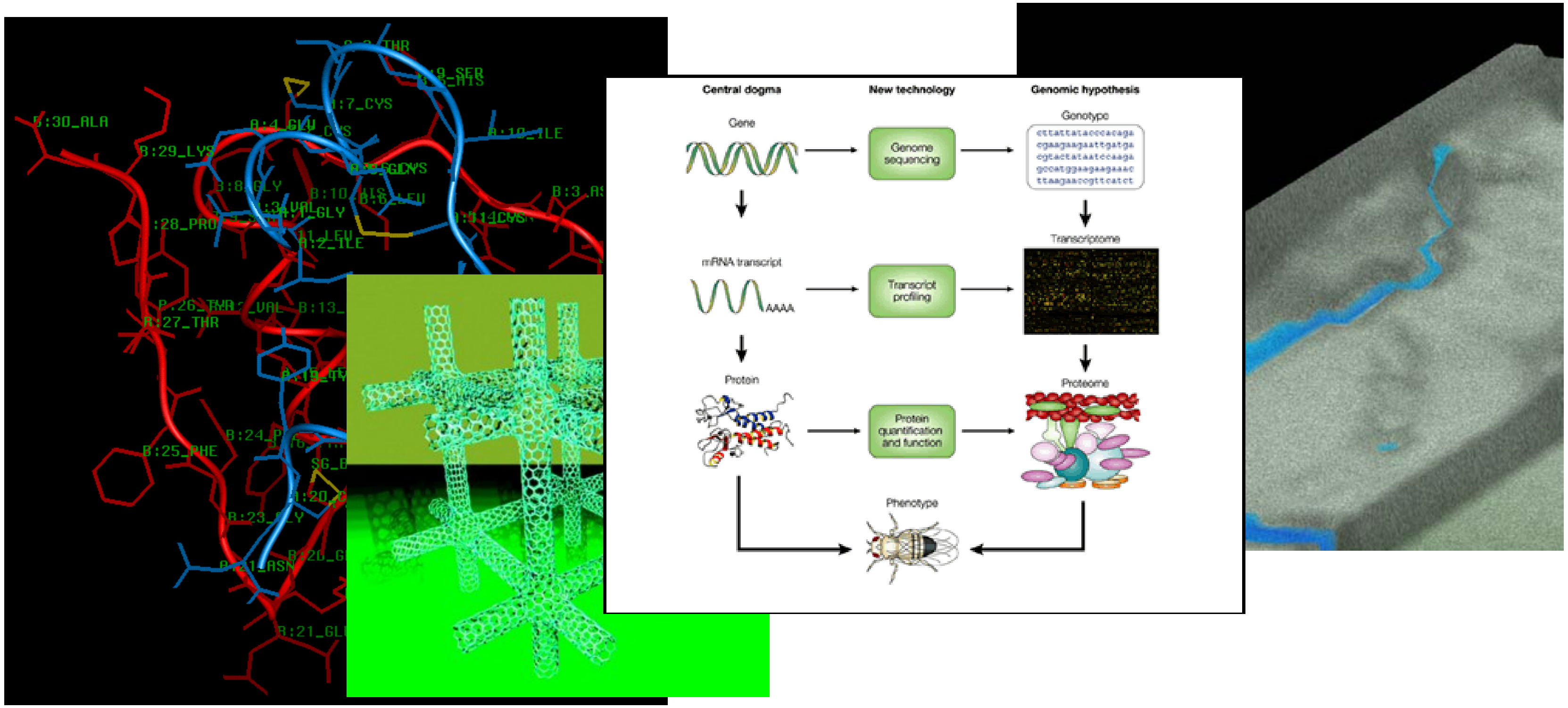


Wind-tunnel Simulation

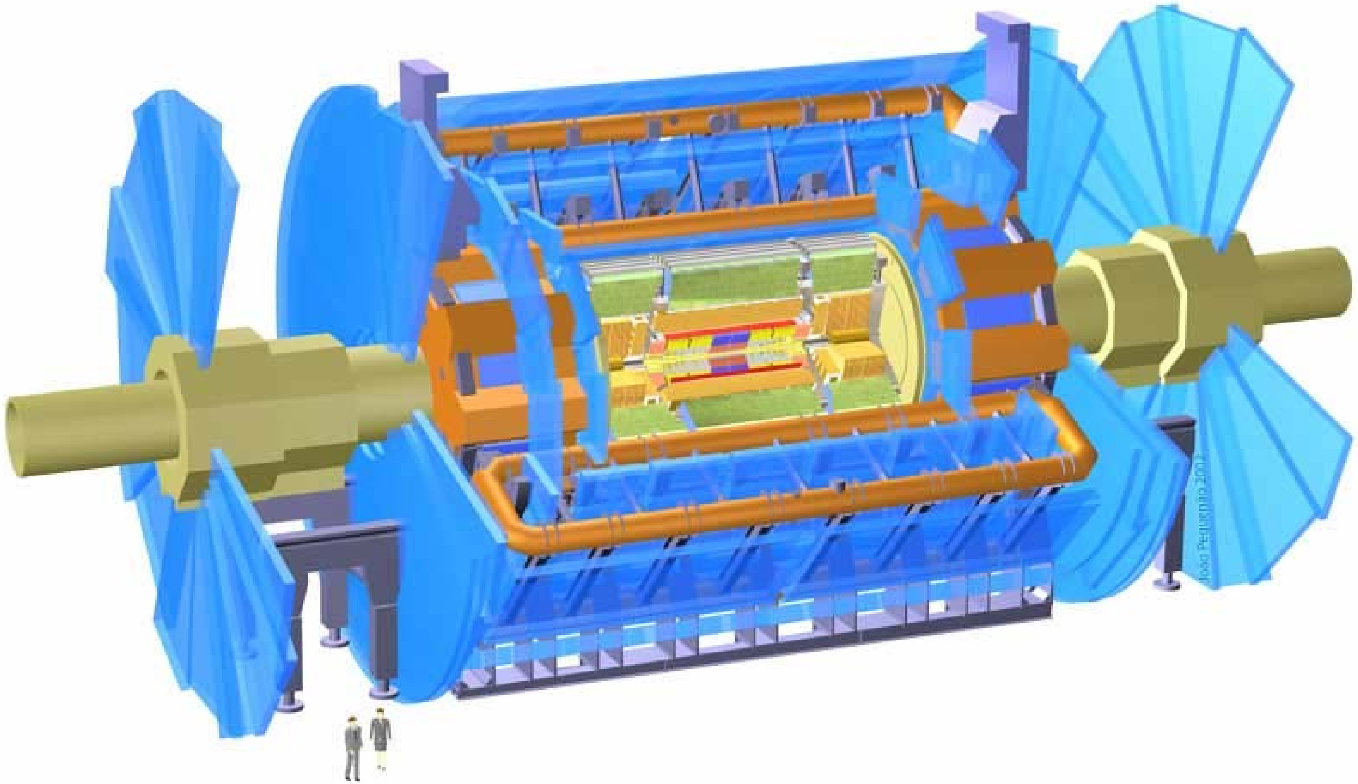


E-science

”Science (increasingly) done through distributed global collaborations enabled by the Internet, using very large data collections, tera-scale computing resources and high performance visualisation.”

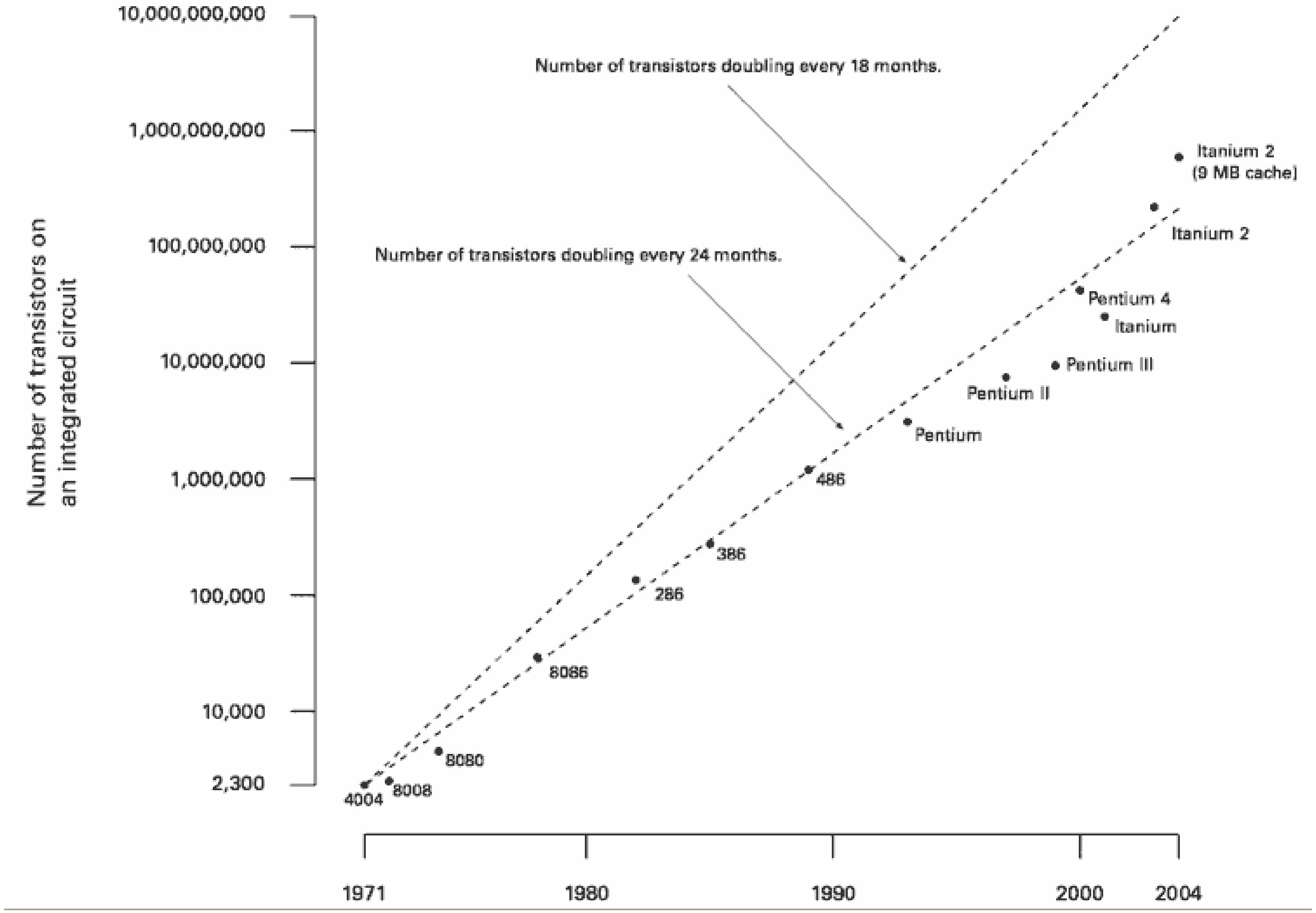


The Atlas experiment



Moore’s Law

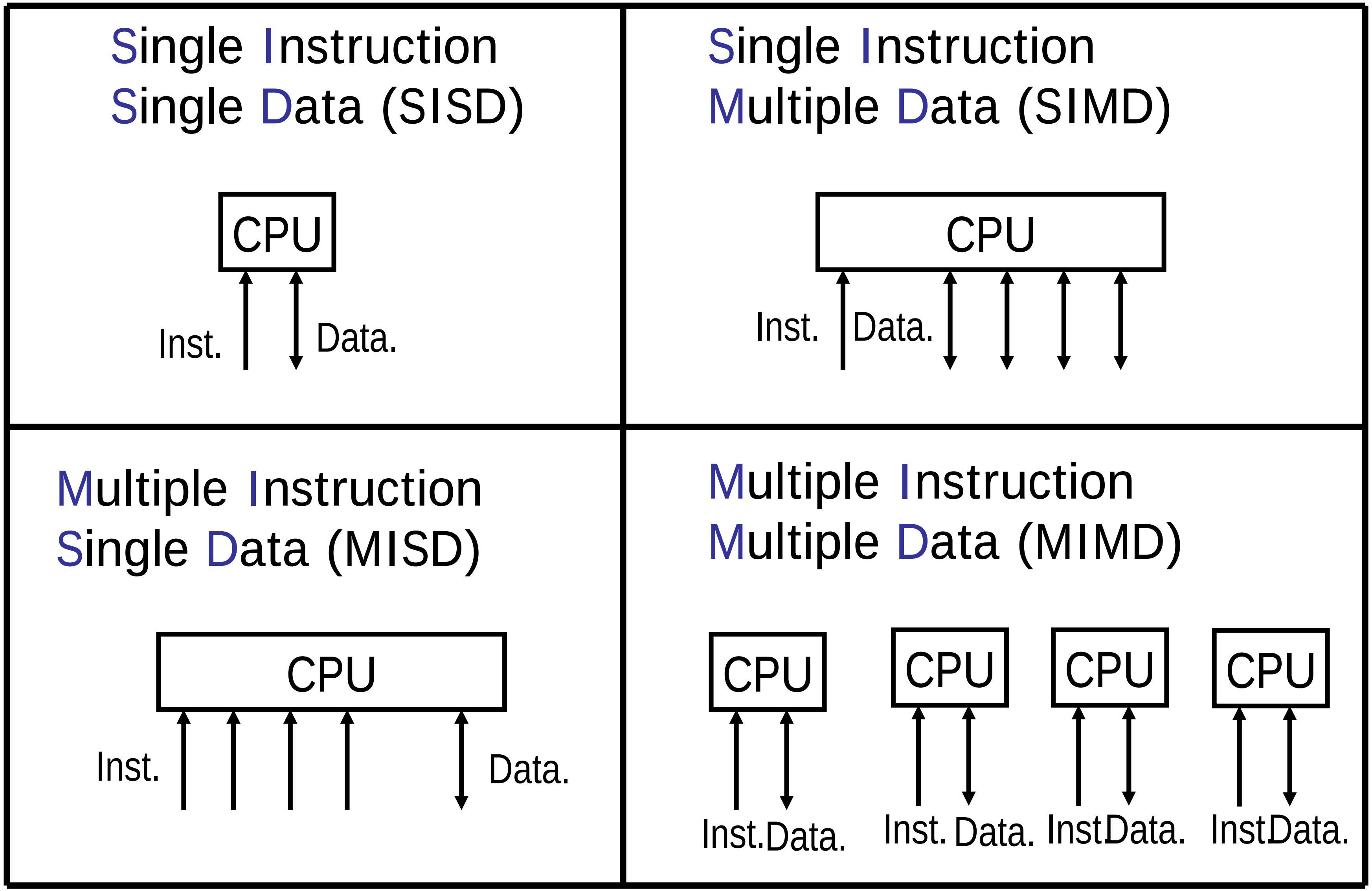
“The number of transistors that can be inexpensively placed on an IC is increasing exponentially, doubling approximately every two years “



Parallel Computing

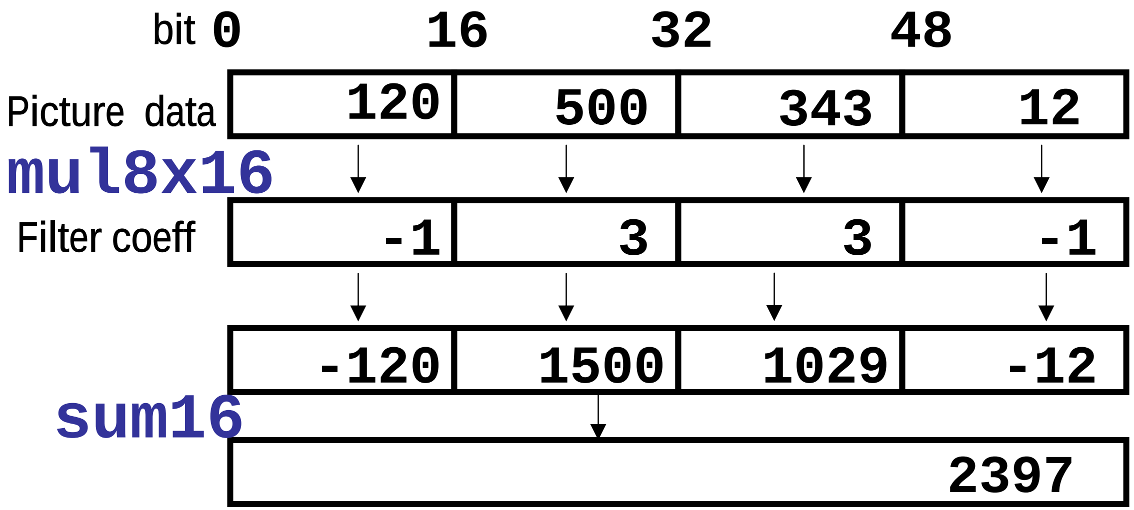
- Using several CPU's to speed up ***computation***
- ***Speedup:*** T(1)/T(n)
- linear speedup desired
- No single CPU is fast enough:
  - Economy (mass production)
  - Switching speed of transistors
  - Speed-of-light argument

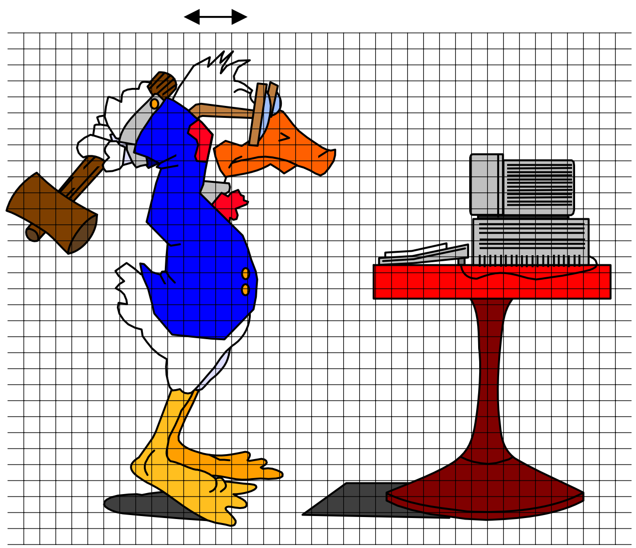
Flynns Taxonomi



SIMD (instruction-level parallelism)

- Single Instruction Multiple data
- VIS, MMX

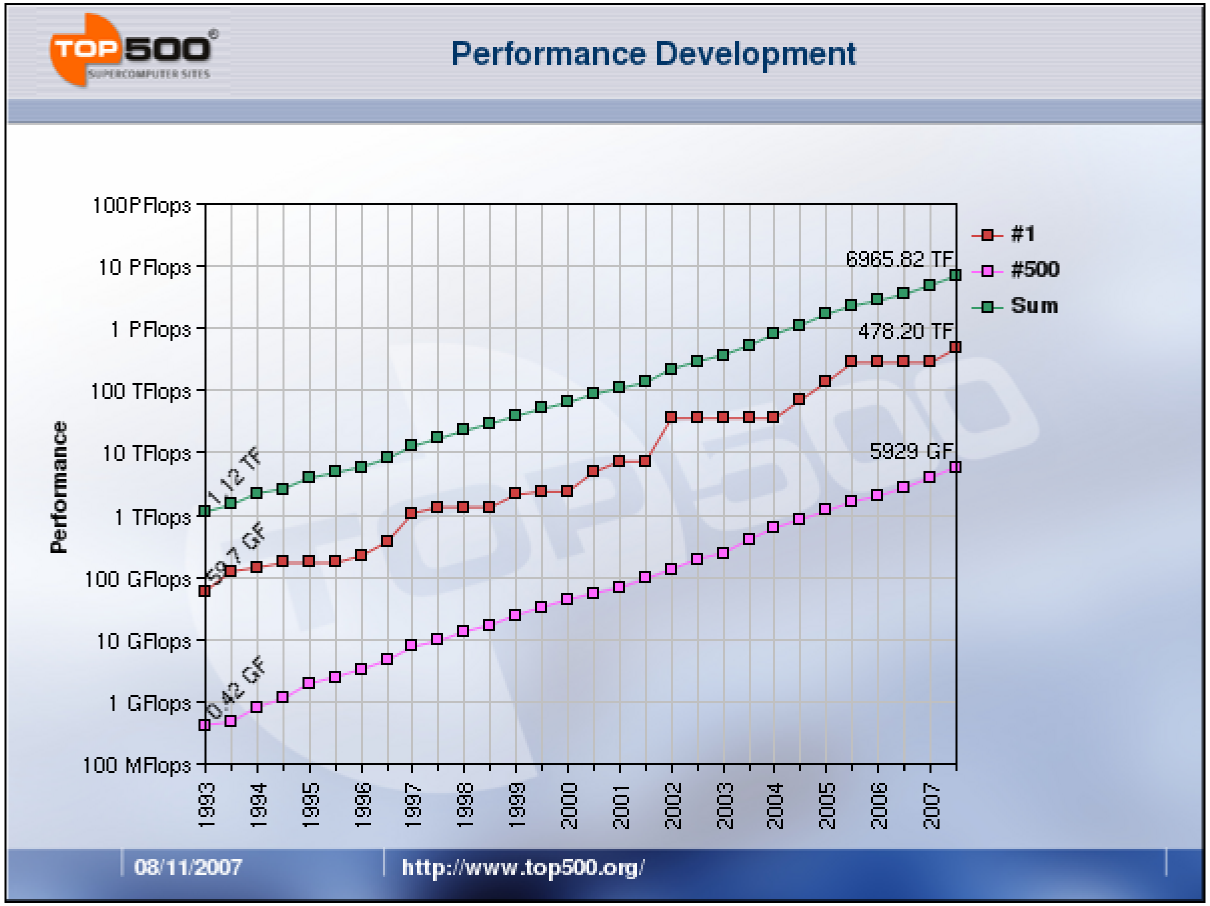




- Vector processors

Development in Compute Power

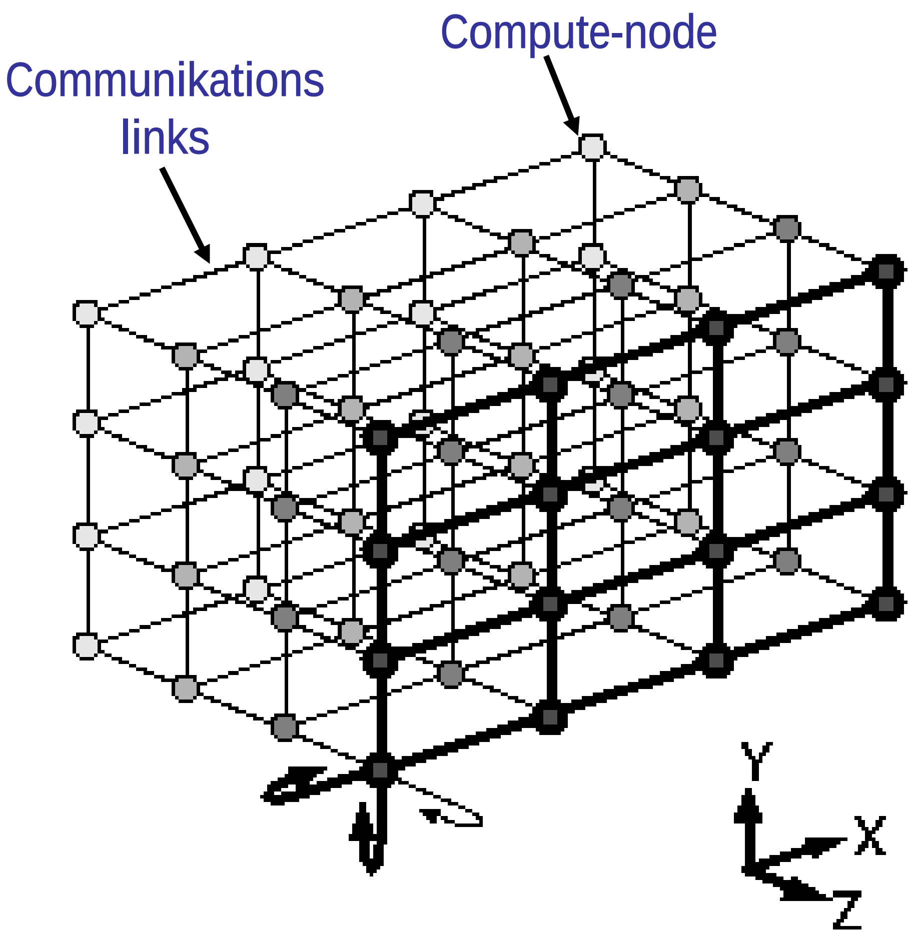
NB: Log scale

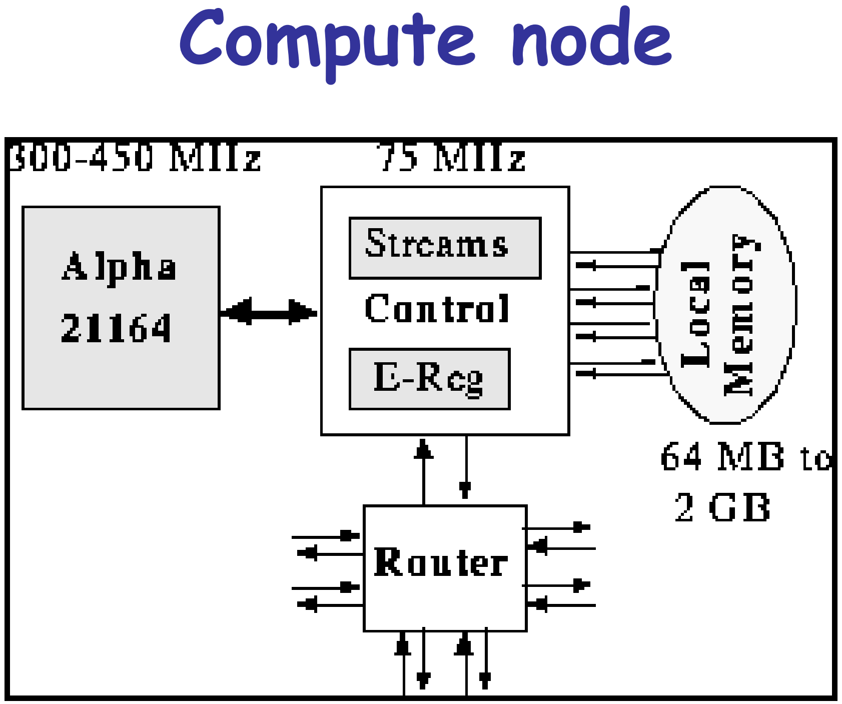


No 1: BlueGene/L System, a joint development of IBM and the Department of Energy's (DOE) National Nuclear Security Administration (NNSA) and installed at DOE's Lawrence Livermore National Laboratory

Cray T3E super computer anno ‘98

T3E Architecture





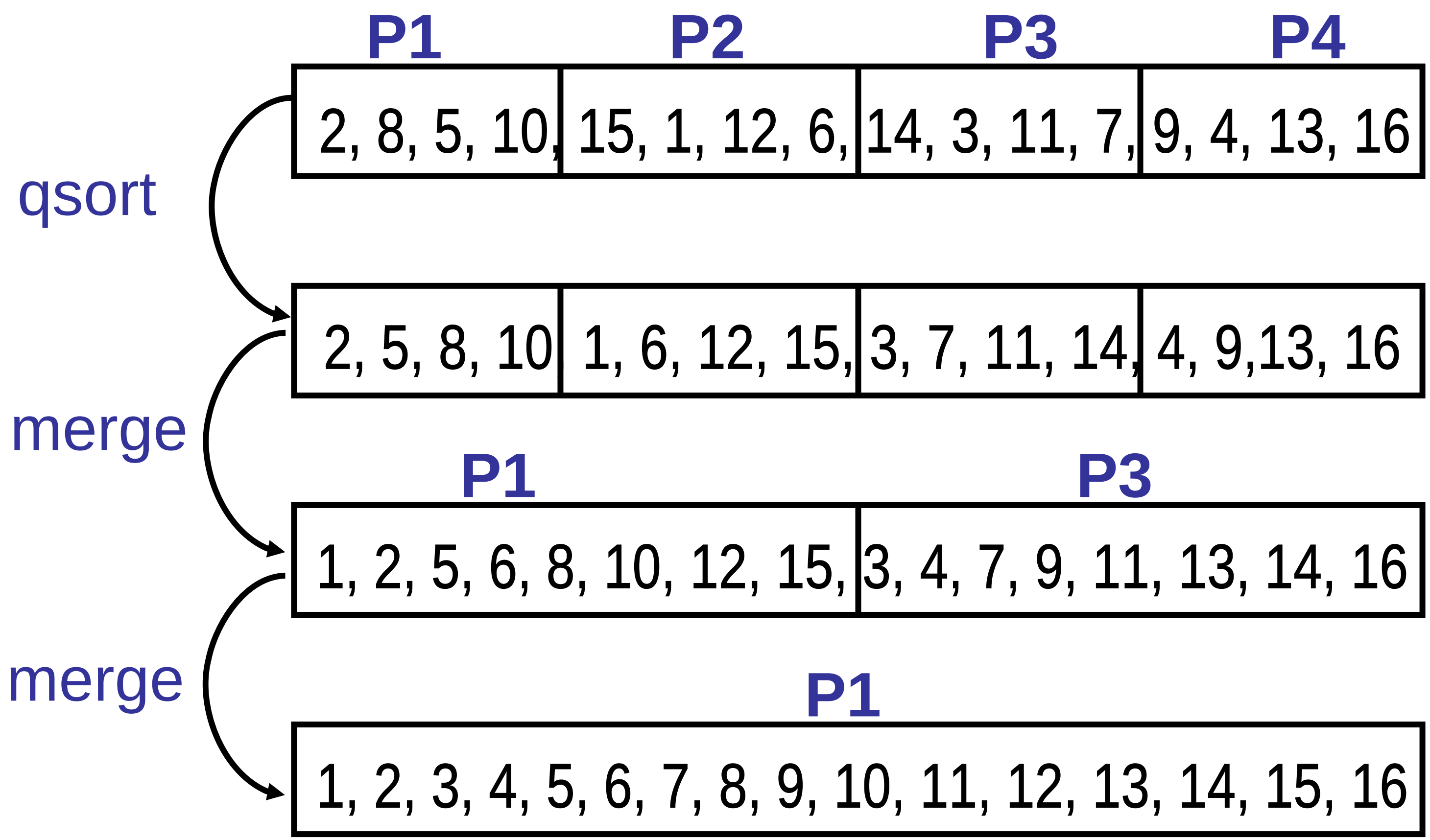
Up to 2048 nodes

New AAU Cluster

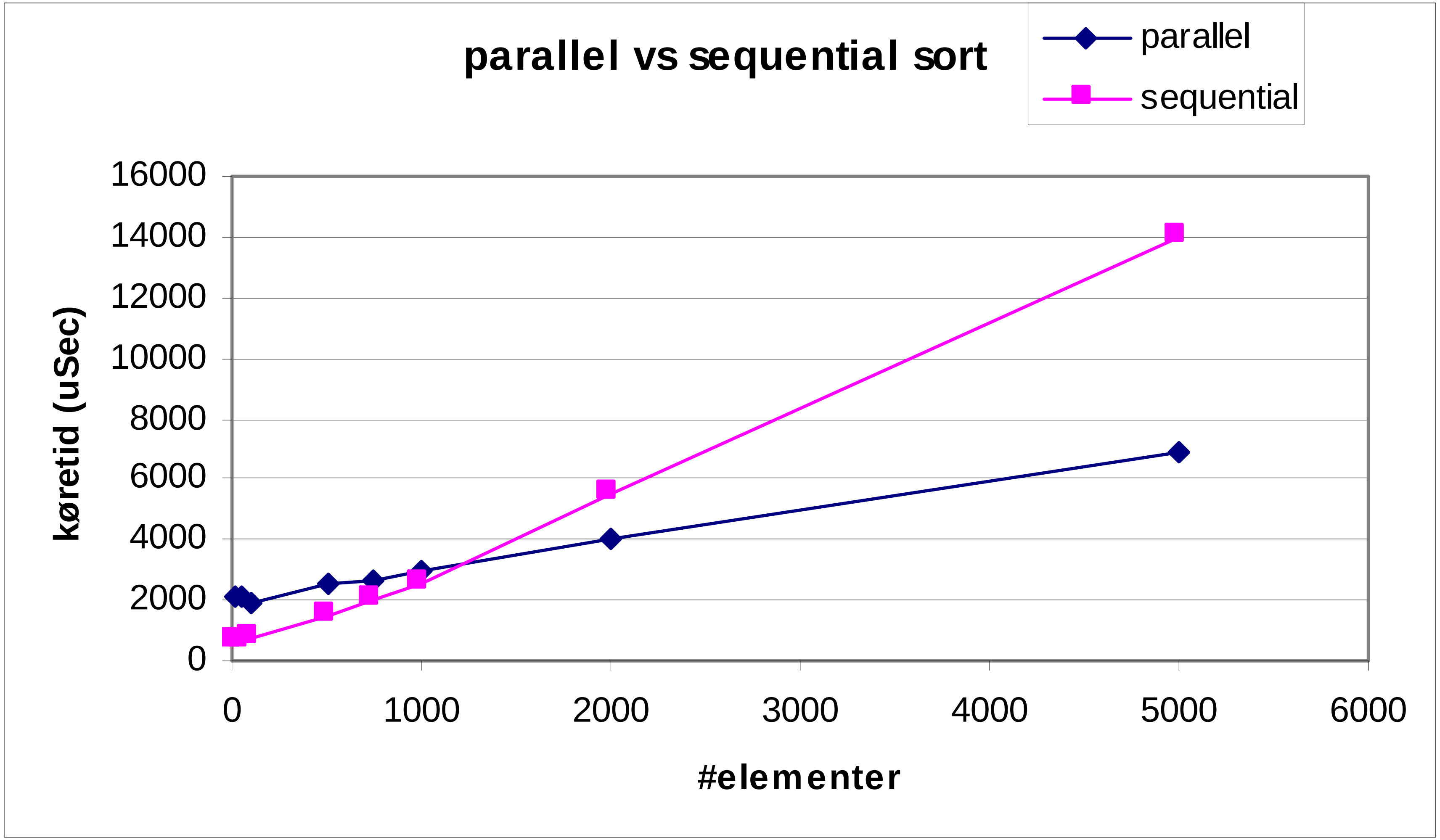
- Theoretical peak performance: 6.6 Tera-flops
- CPU Cores: 672
  - Intel quad core-2 cpu's at 2.33 MHz
  - Dual cpu config per blade
- Main memory: 1344 GB
- Interconnect:
  - GBit ethernet, and
  - Infiniband (20 Gbit/s, low latency)
- Storage: 4.5 TB
- Power consumption: 33KW
- Occupies 2 complete rack stands

- Cooling costs = 1/3 of HW costs in a 4 year period

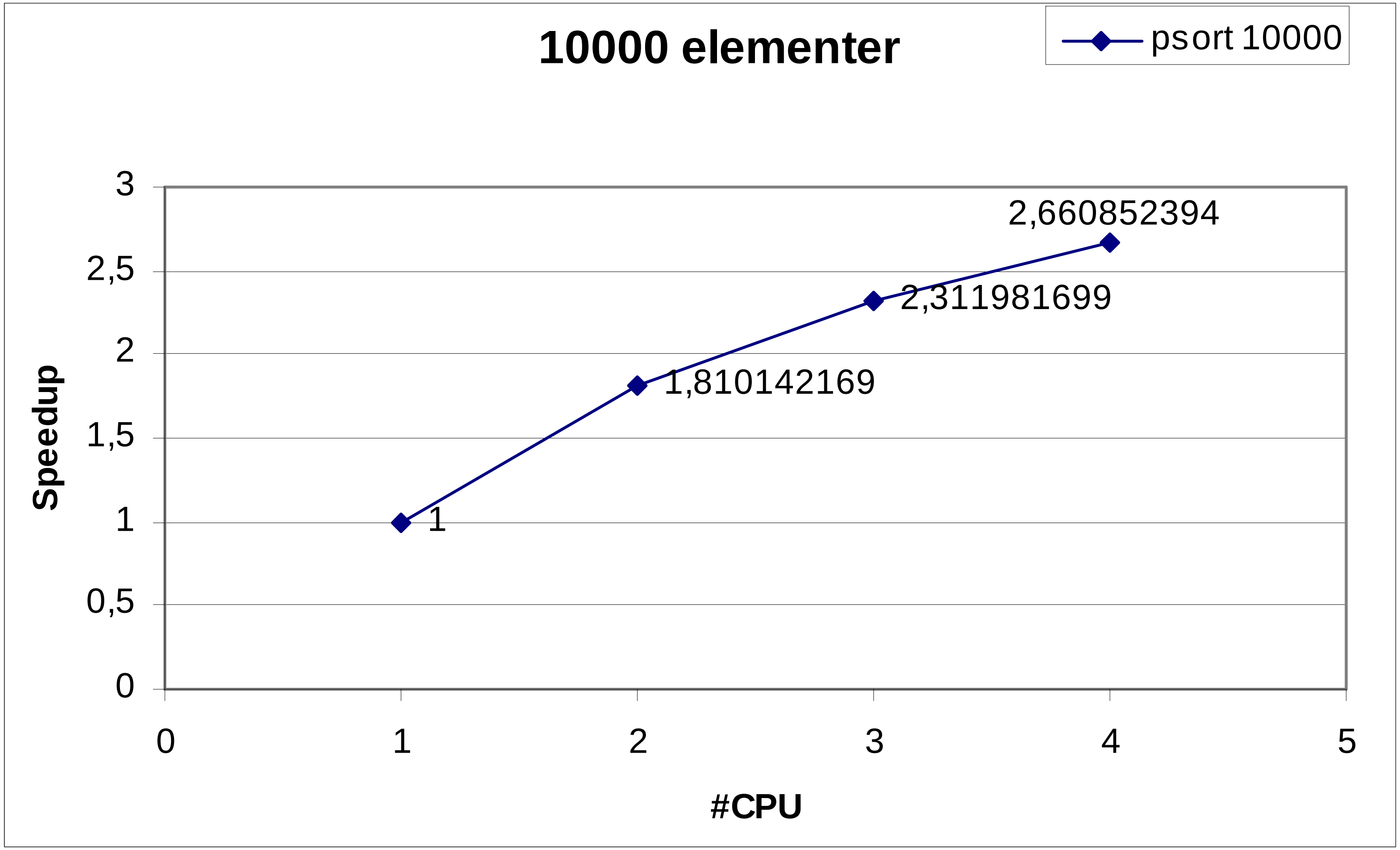
Simpel Parallel Sorting



Gain of using parallelism



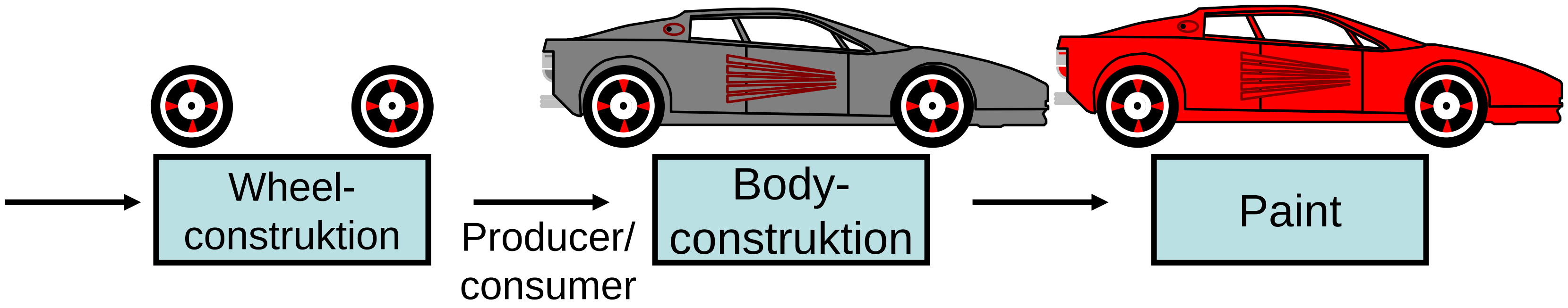
Speedup


10000 elementer
psort 10000
Speedup
3
2,5
2
1,5
1
0,5
0
0
1
2
3
4
5
#CPU
1
1,810142169
2,311981699
2,660852394

Principles

- Pipelining
- Instruktion level parallelism
  - pipelining
  - SIMD
    - Vector processing
    - MMX, VIS
- Data-flow
- SMP's
- Multi-computers, Clusters
- Explicit vs. parallelizing compilers

Pipelining



- T=Assembly time per finished product
- N stations
- Production rate: T/N

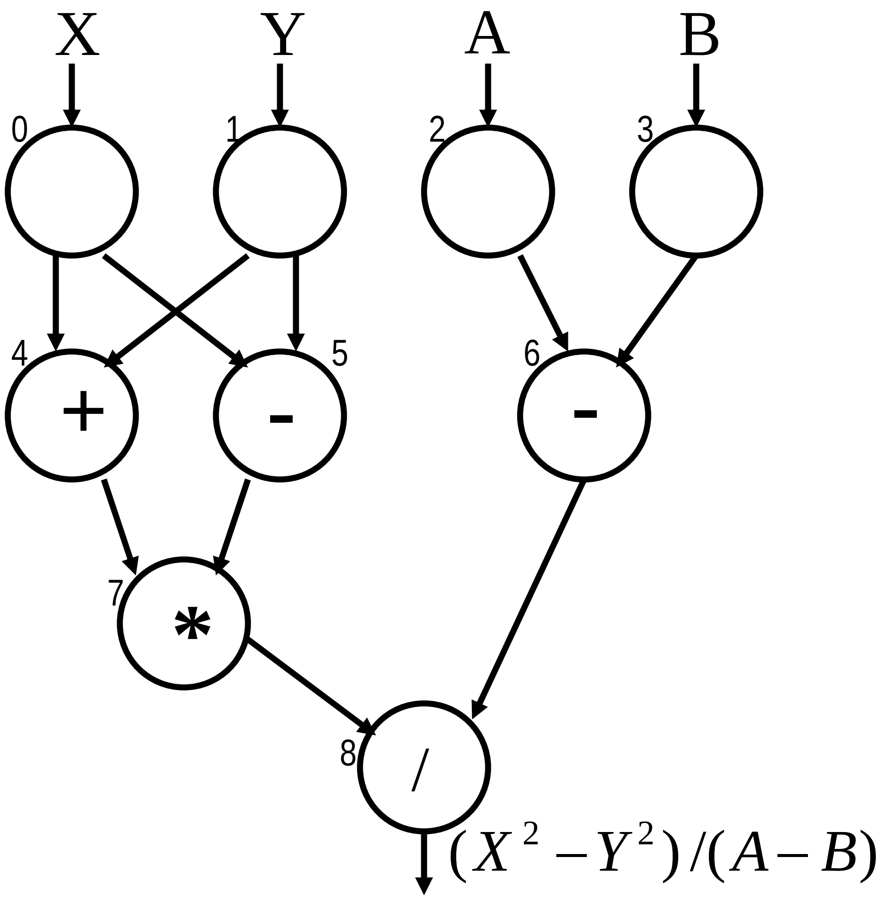
Dataflow

Dataflow

```
0: In 4 L , 5 L -- Read X
1: In 4 R , 5 R -- Read Y
2: In 6 L       -- Read A
3: In 6         -- Read B
4: + 7 L        -- (X+Y)
5: - 7 R        -- (X - Y)
6: + 8 R        -- (A + B)
7: * 8 L        -- (X+Y)*(X-Y)
8: / , Out
```

Conventional

```
0: Load R1, X
1: Load R2, Y
2: Load R3, A
3: Load R4, B
4: + R1, R2 , R5(R5 = R1 + R2)
5: - R1, R2, R6 (R6 = R1 - R2)
6: + R3, R4, R7 (R7 = R3 + R4)
7: * R5, R6, R8 (R8 = R5 * R6)
8: / R8, R7, R9 (R9 = R8 / R9)
9: Store R9
```



Granularity

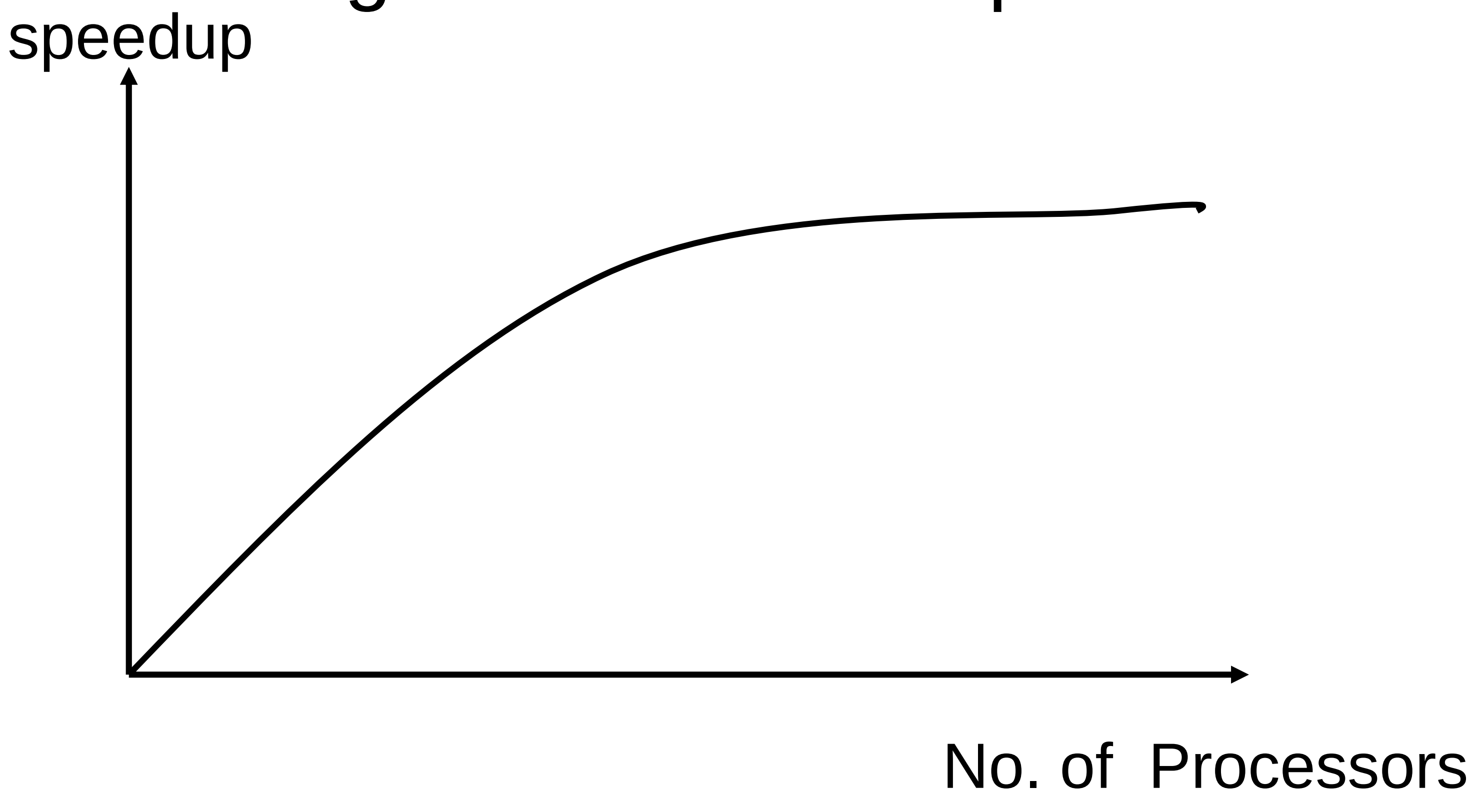
Grain size	Description	Synchronization inverval #instruktions	Exsample
Fine	Parallelism within instruktionsstream	<200	(A*b) + (c+d)
Medium	Multi-threading i n same application	-2000	CFD, Søg +sort
Coarse	Multi-processing in multiprog. OS	-20000	Matrix Multiplication
Very Coarse	Distributed Computation over LAN	-10M	ParMake Mandelbrot
Independen t	Processes from different users	(N/A)	Distr. OS

Obstacles for linear speedup

- Some problems are difficult to parallelize
  - Data dependencies
  - Different computation time for sub-problems
  - Unparallizable sequential code (Amdahls Law)
- Parallelism overhead
  - Creation/scheduling of processes/threads
  - communication
  - Synchronization
- Grain Size
  - Too small grain-size is outweigted by overhead
- Bottlenecks
  - data bus, communication channels, i/o

Speedup

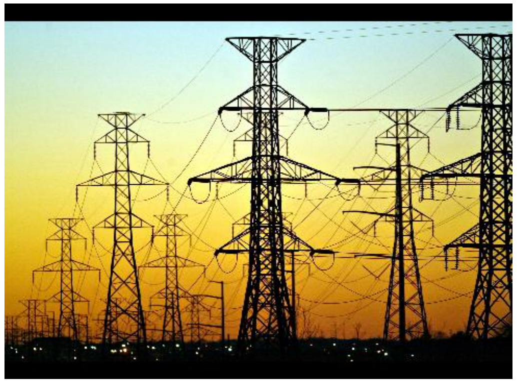
- Linear speedup rarely possible
- Problem: gain as much as possible



Grid Computing

The grand vision

A huge virtual distributed computer.



GRID-Definition

“A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.”

The Grid – a blueprint for a new computing infrastructure, 1998

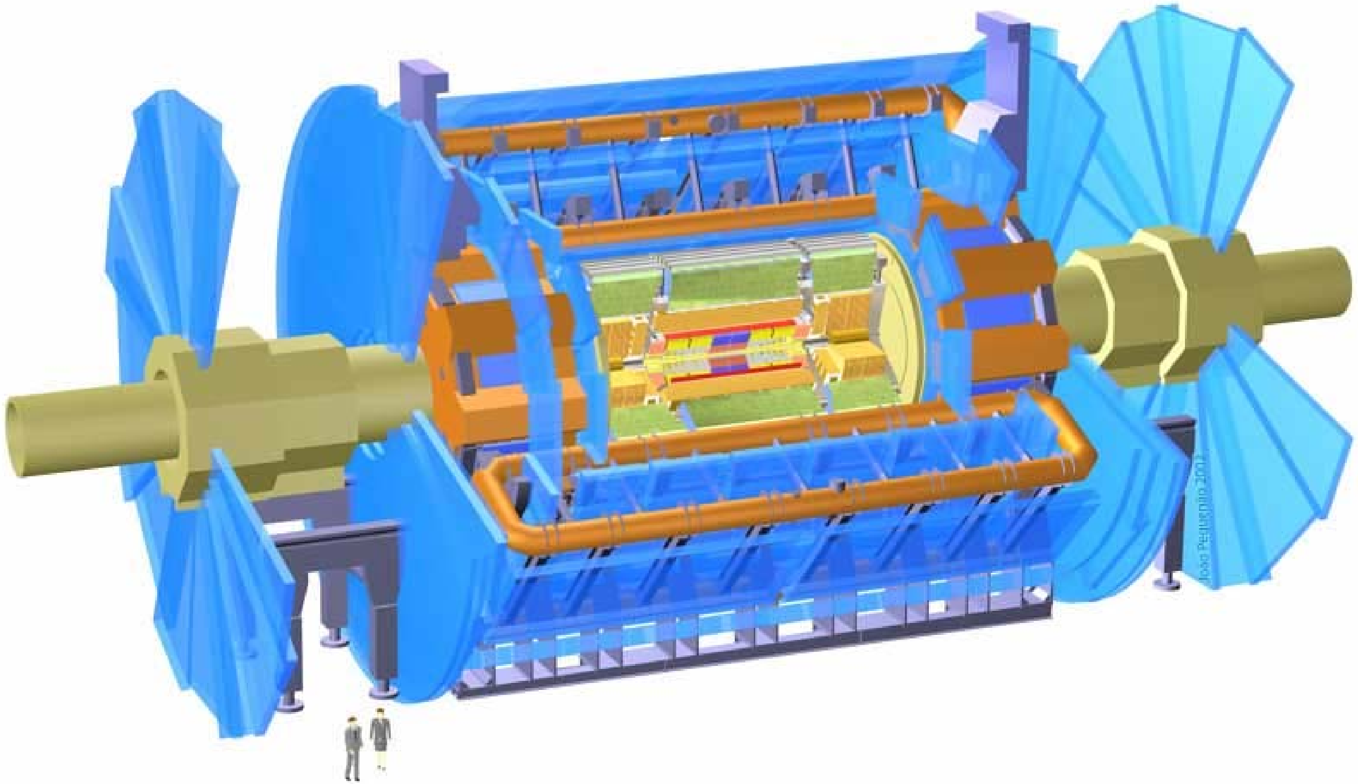
Challenges

- Make hardware owned by *different* organizations available to *non-members* of that organization.
- In such a way that normal operation of the equipment can continue.
- In such a way that the organization still can control who gets access.
- In such a way that we can control who gets access to specific pieces of data.
- In such a way that operations can be performed anonymously.
- And still charge for the use of hard- and software.

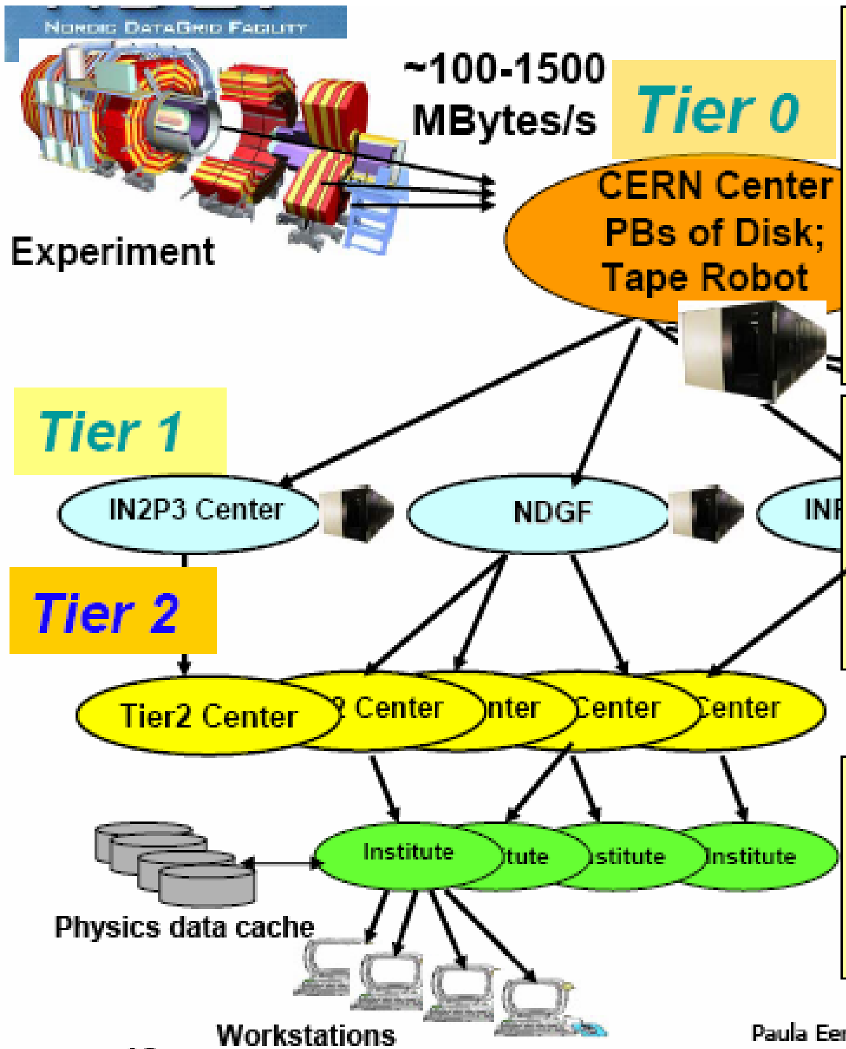
Challenges

- Resource allocation and scheduling
- Authentication and authorization
- Protection
- Control
- Accounting

The Atlas experiment



LHC Computing Hierarchy

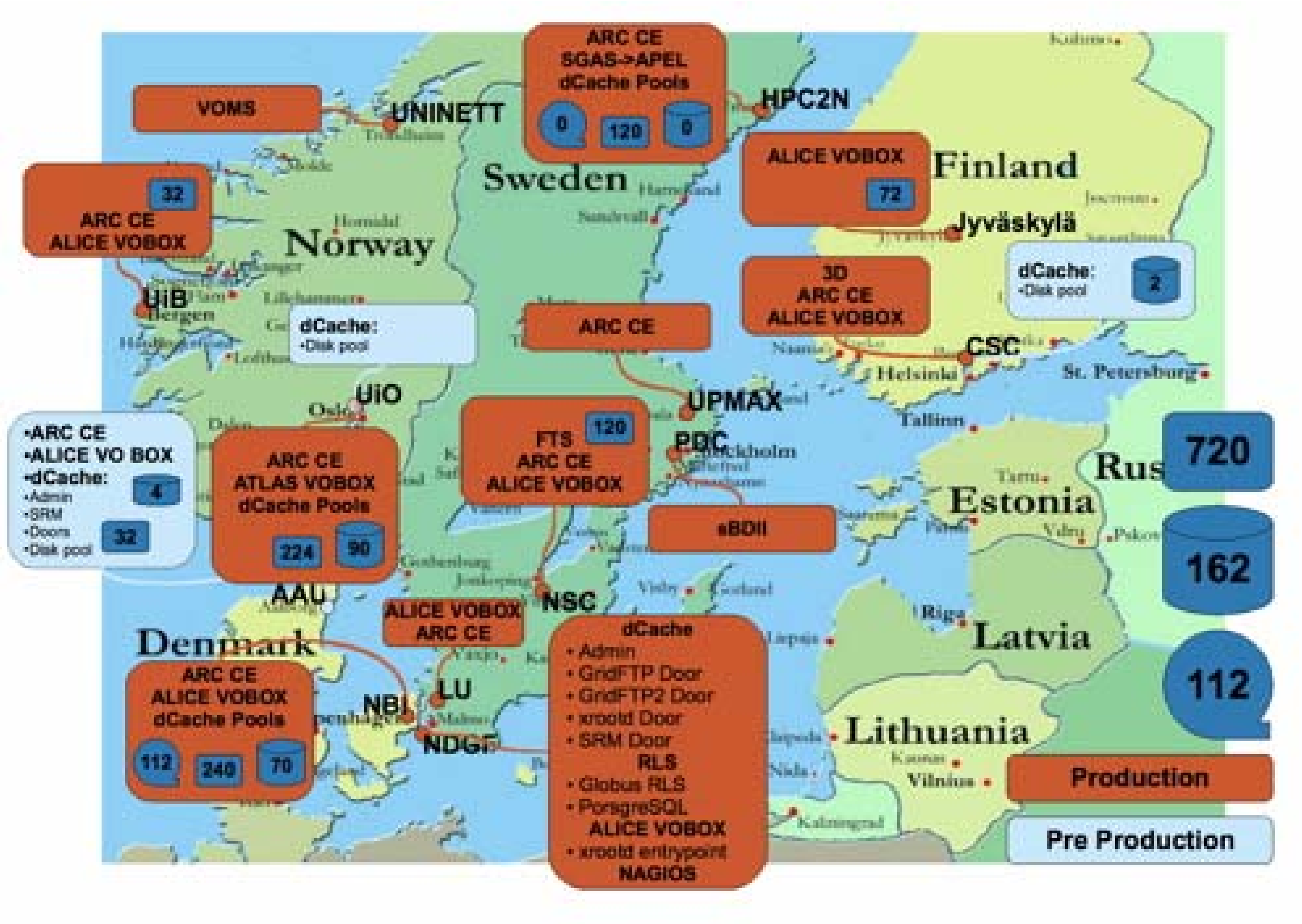


Tier 0= CERN. Tier 0 receives raw data from the Experiments and records them on permanent mass storage. 5-8 PetaBytes of data/year, can grow up to 100 PB/year. First-pass reconstruction of the data, producing summary data.

Tier 1 Centres = large computer centres (12). Tier 1's provide permanent storage and management of *raw*, *summary* and other data needed during the analysis process.

Tier 2 Centres = smaller computer centres (several 10's). Tier 2 Centres provide disk storage and concentrate on simulation and end-user analysis.

Nordic Data Grid Facility

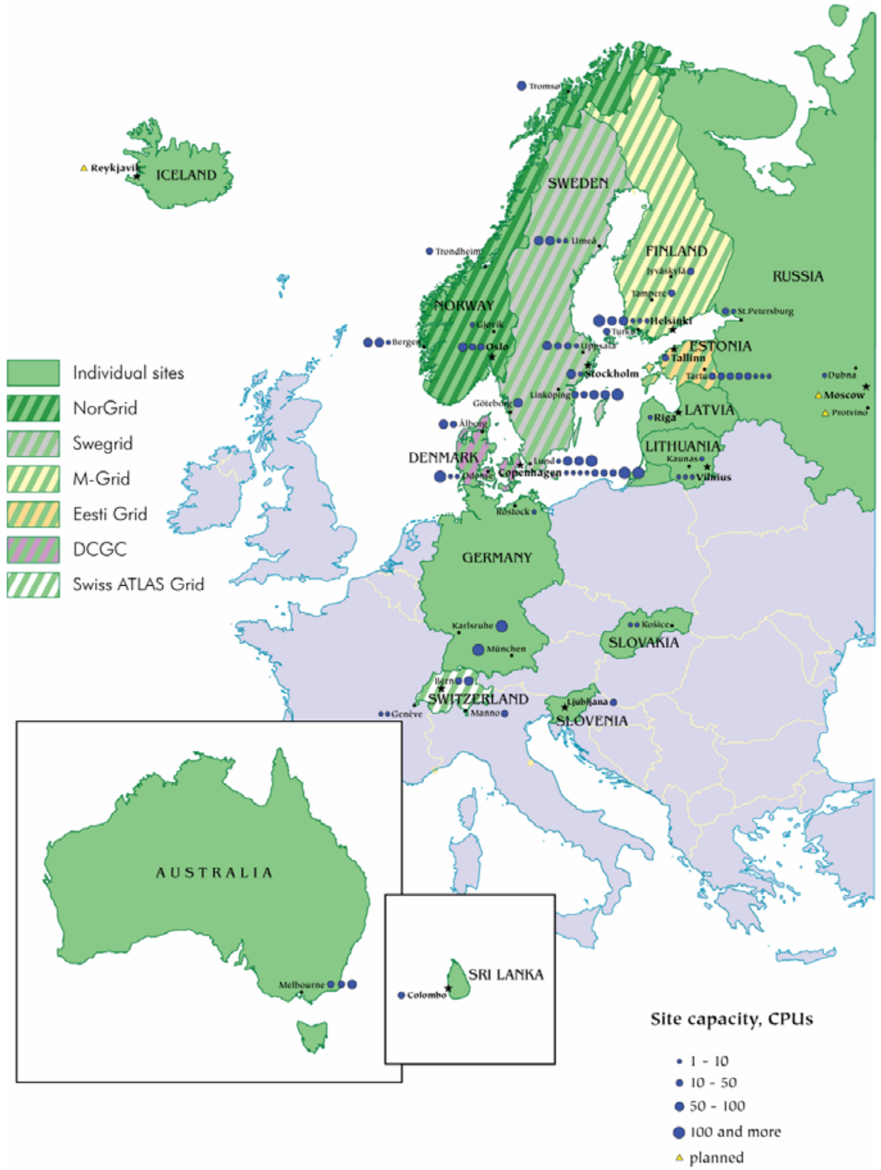


NorduGrid

- NorduGrid is a collaboration between a number of universities mostly located in the Nordic countries.
- NorduGrid Advanced Resource Connector is:
  - A Globus-based Grid middleware solution
- NorduGrid is a production Grid ('05)
  - Approximately 5000 CPUs
  - Approximately 75 TB of storage

Web: www.nordugrid.org

The Nordic Grid



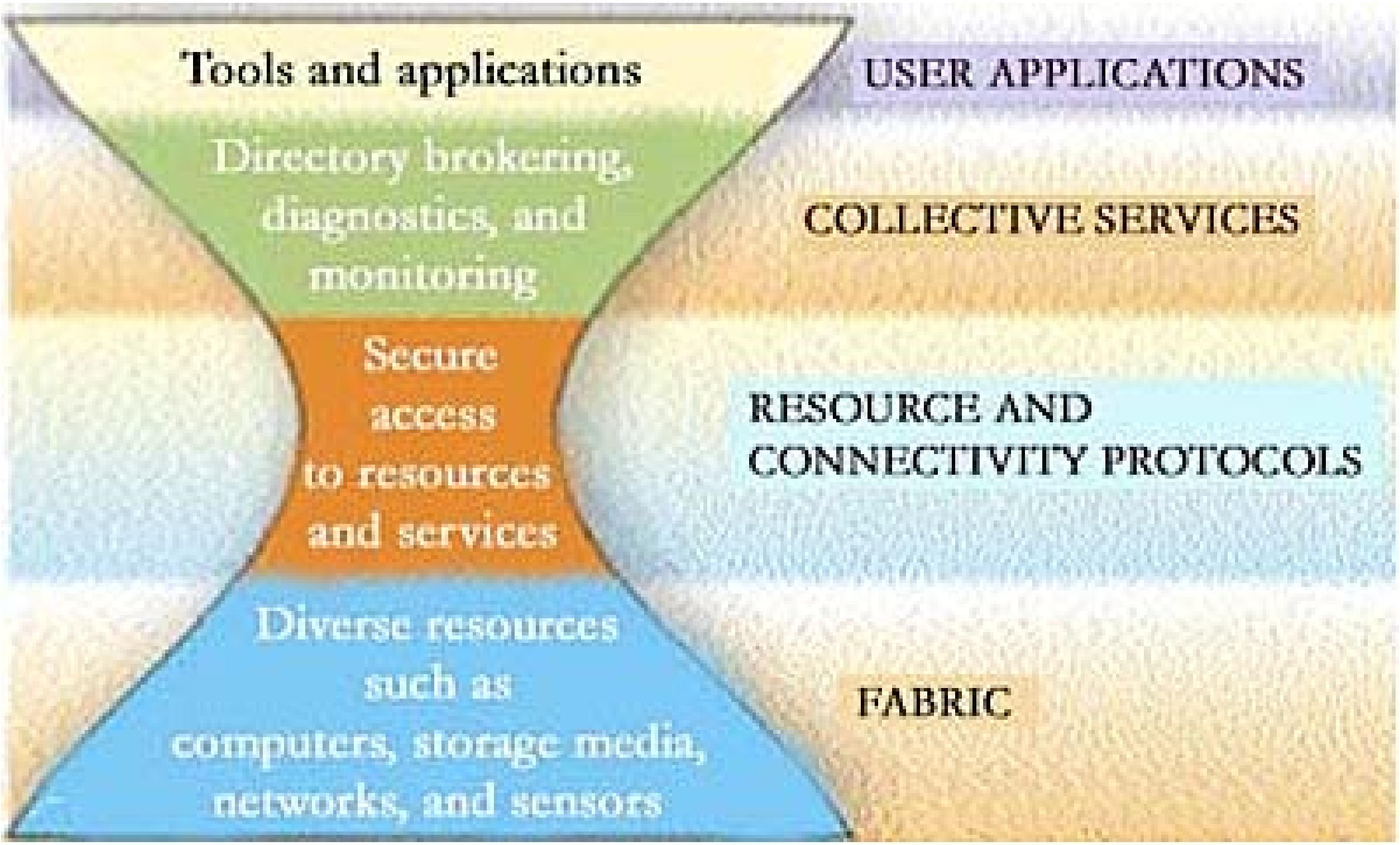
Globus

the globus alliance

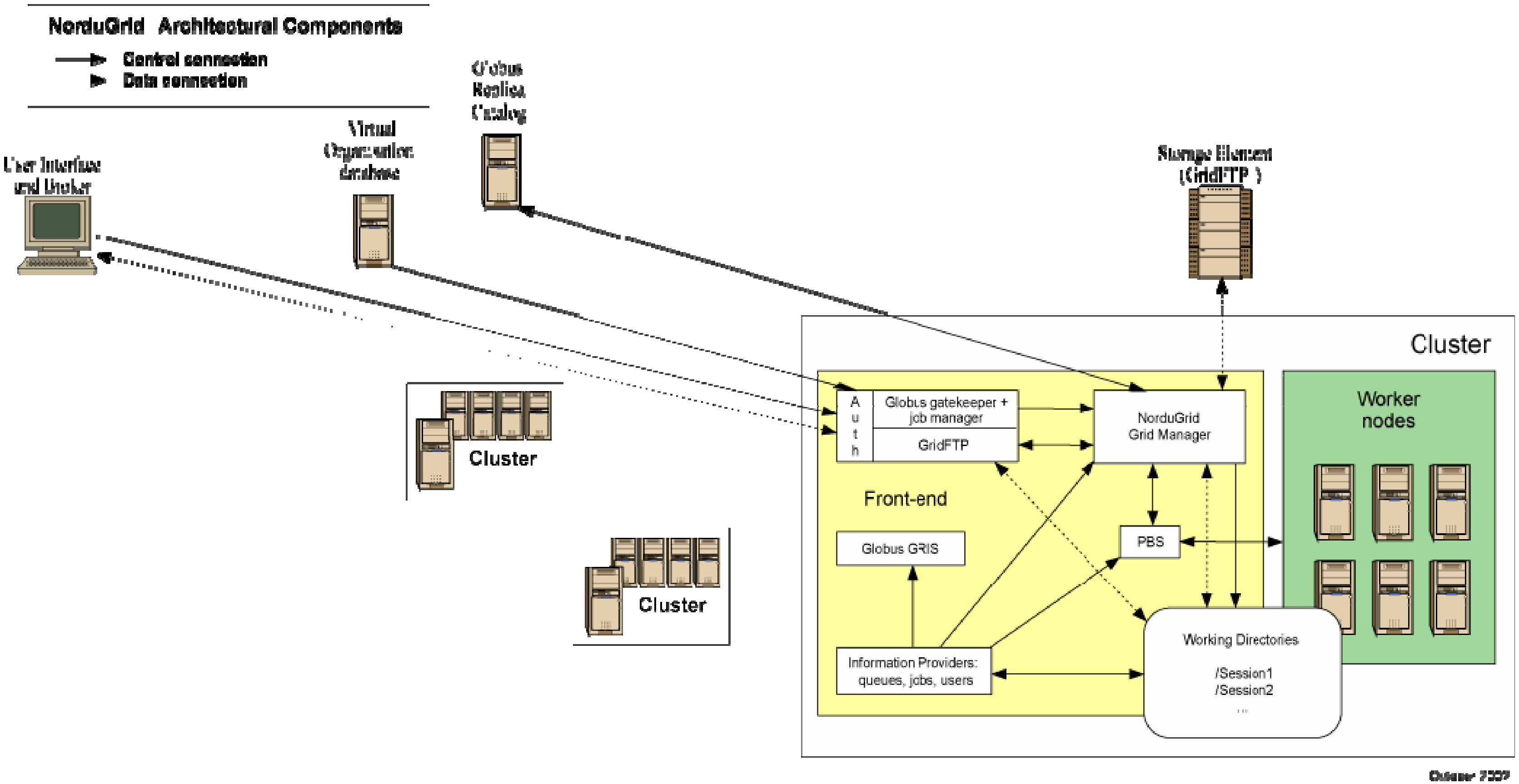
- An open source software toolkit used for building grids.
- Includes software services and libraries for resource monitoring, discovery, and management, plus security and file management.

Web: www.globus.org

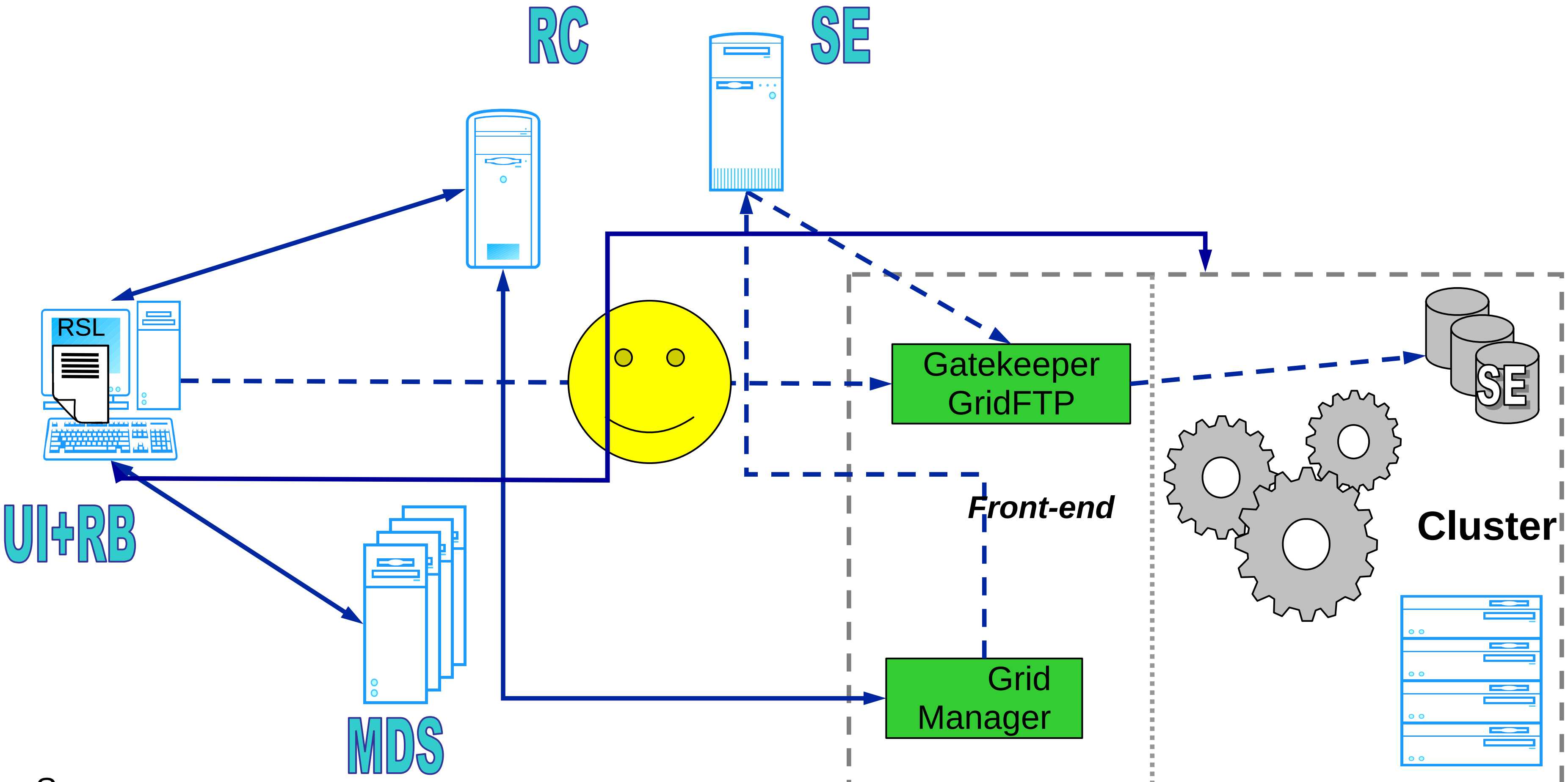
The globus model



System Components (ARC)

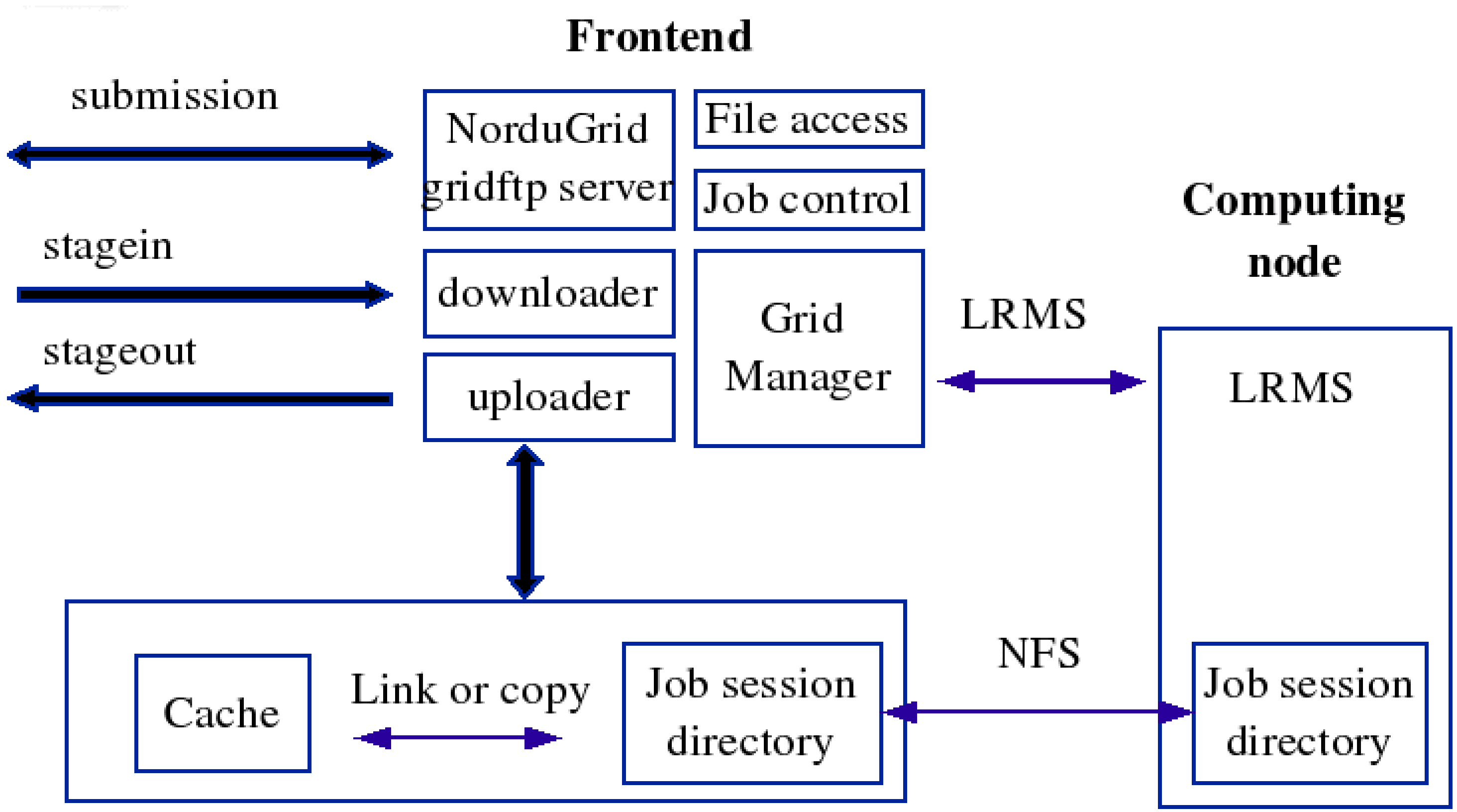


Workflow



Source
NorduGrid.org

Front-end


Frontend
submission
stagein
stageout
NorduGrid gridftp server
File access
Job control
downloader
uploader
Grid Manager
LRMS
Computing node
LRMS
Cache
Link or copy
Job session directory
NFS
Job session directory

The user-interface

ngsub	to submit a task
ngstat	to obtain the status of jobs and clusters
ngcat	to display the stdout or stderr of a running job
ngget	to retrieve the result from a finished job
ngkill	to cancel a job request
ngclean	to delete a job from a remote cluster
ngrenew	to renew user’s proxy
ngsync	to synchronize the local job info with the MDS
ngcopy	to transfer files to, from and between clusters
ngremove	to remove files

Broker

- The user must be authorized to use the cluster and the queue
- The cluster's and queue's characteristics must match the requirements specified in the xRSL string (max CPU time, required free disk space, installed software etc)
- If the job requires a file that is registered in a Replica Catalog, the brokering gives priority to clusters where a copy of the file is already present
- From all queues that fulfills the criteria one is chosen randomly, with a weight proportional to the number of free CPUs available for the user in each queue
- If there are no available CPUs in any of the queues, the job is submitted to the queue with the lowest number of queued job per processor