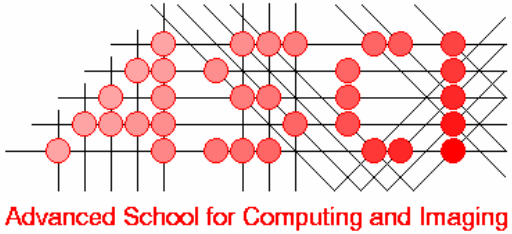
Distributed supercomputing on DAS, GridLab, and Grid'5000

Henri Bal

Vrije Universiteit Amsterdam
Faculty of Sciences

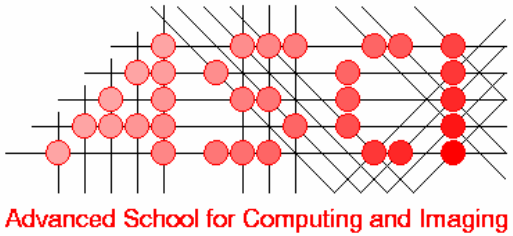


Introduction

- **DAS (Distributed ASCI Supercomputer) has a long history and continuity**
  - DAS-1 (1997), DAS-2 (2002), DAS-3 (July 2006?)
- **Simple *Computer Science* grid that *works***
  - Over 200 users, 25 Ph.D. theses
  - Stimulated new lines of CS research
  - Used in international experiments
- **Colorful future: DAS-3 is going optical**

Outline

- **History**
- **Impact on Dutch computer science research**
  - Trend: cluster computing ➔ distributed computing ➔ Grids ➔ Virtual laboratories
- **Example research projects**
  - Ibis, Satin
- **Grid experiments on DAS-2, GridLab, Grid'5000**
- **Future:** DAS-3



Organization

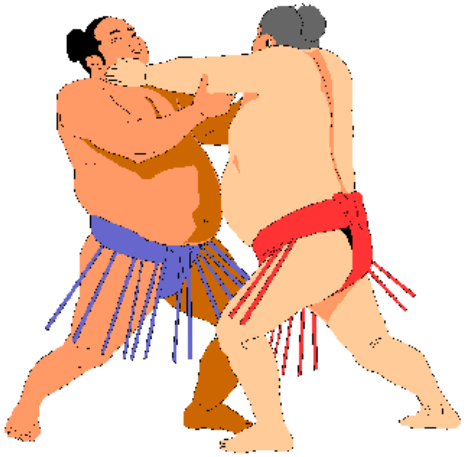
- **Research schools (Dutch product from 1990s)**
  - Stimulate top research & collaboration
  - Organize Ph.D. education
- **ASCI:**
  - Advanced School for Computing and Imaging (1995-)
  - About 100 staff and 100 Ph.D. students
- **DAS proposals written by ASCI committees**
  - Chaired by Tanenbaum (DAS-1), Bal (DAS-2, DAS-3)

Design philosophy

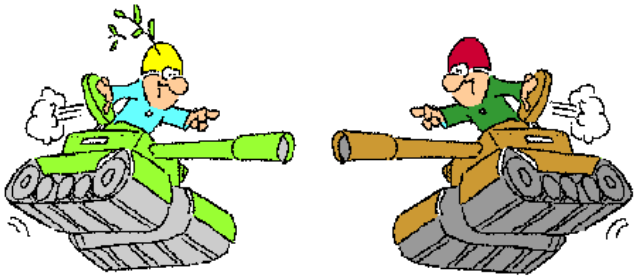
- **Goals of DAS-1 and DAS-2:**
  - Ease collaboration within ASCI
  - Ease software exchange
  - Ease systems management
  - Ease experimentation
- **➔ Want a clean, laboratory-like system**
- **Keep DAS simple and *homogeneous***
  - Same OS, local network, CPU type everywhere
  - Single (replicated) user account file

Behind the screens ….

Artist's Rendition of the First OS Discussion



Artist's Rendition of the Second OS Discussion

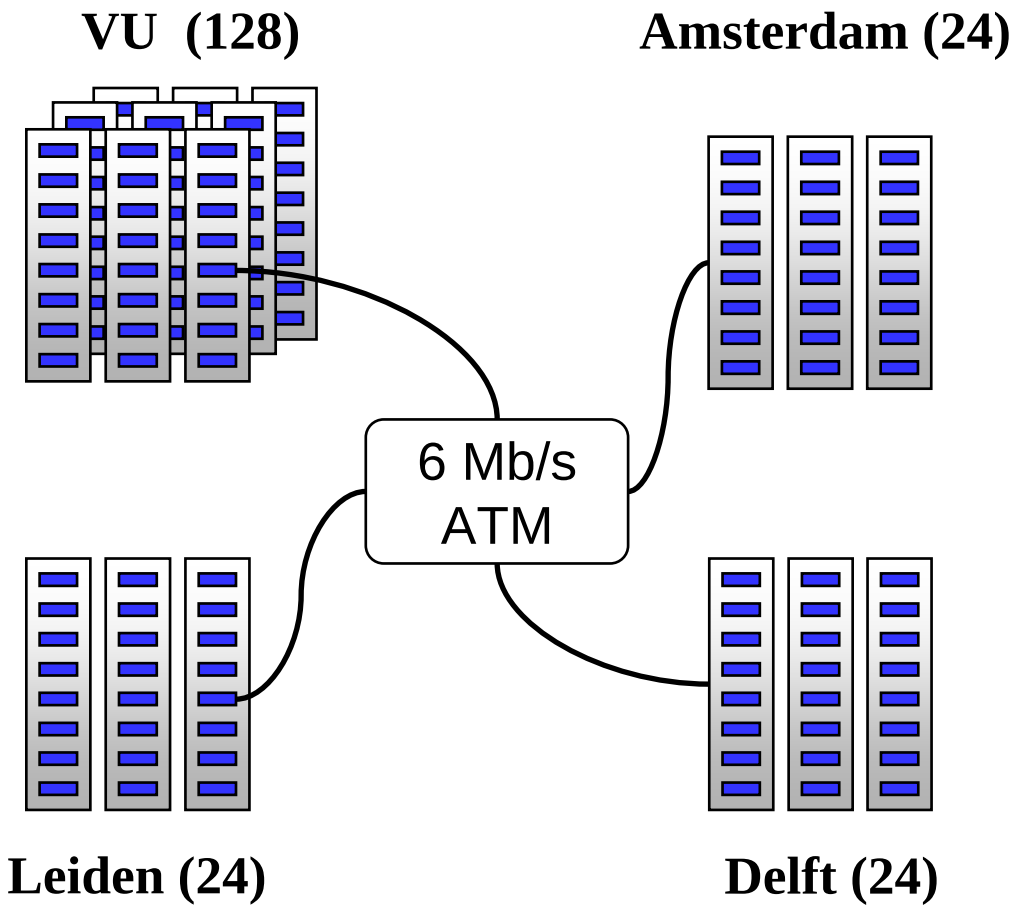


Source: Tanenbaum (ASCI'97 conference)

DAS-1 (1997-2002)

Configuration

200 MHz Pentium Pro
Myrinet interconnect
BSDI => Redhat Linux

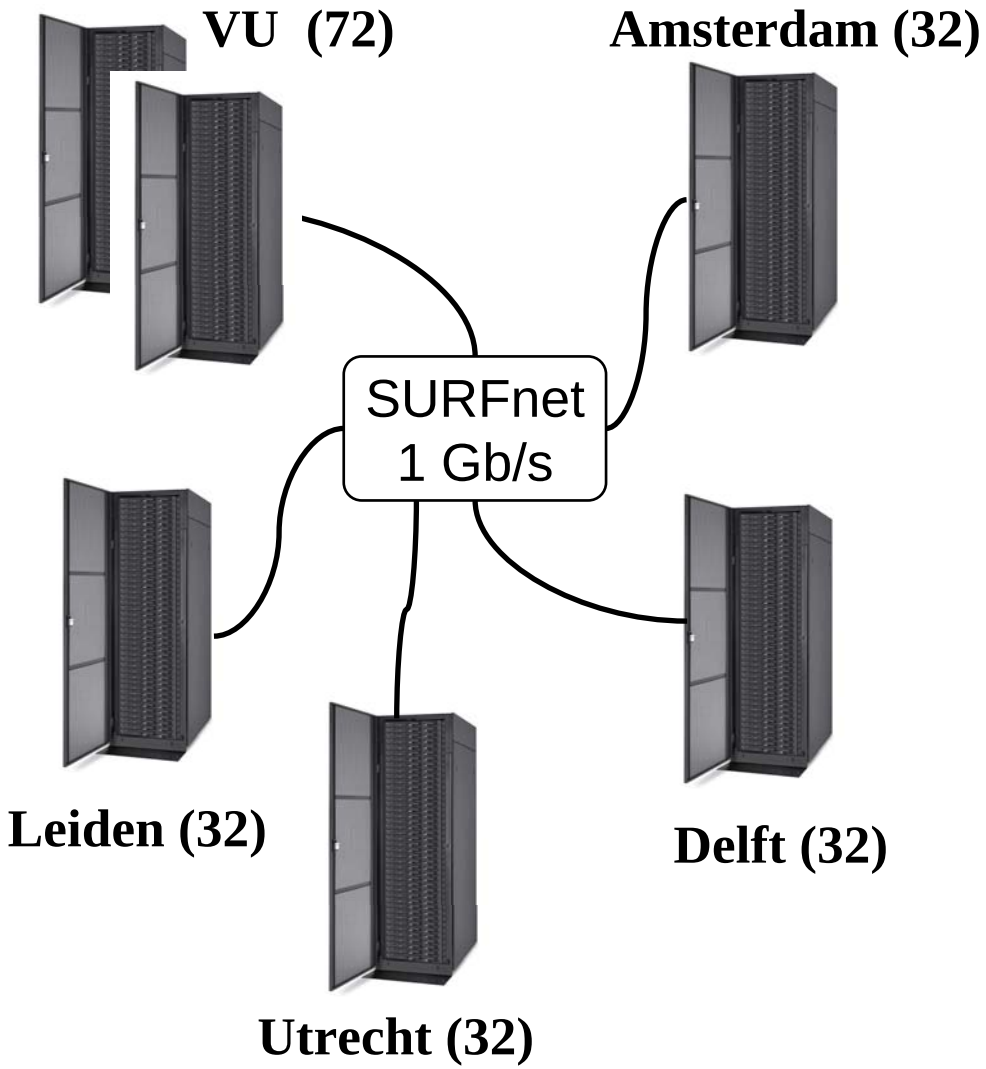


DAS-2 (2002-now)

Configuration

two 1 GHz Pentium-3s
>= 1 GB memory
20-80 GB disk

Myrinet interconnect
Redhat Enterprise Linux
Globus 3.2
PBS => Sun Grid Engine

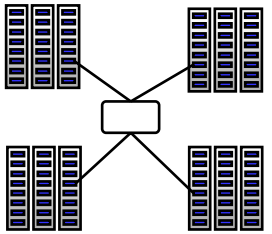


DAS accelerated research trend

Cluster computing

↓

Distributed computing

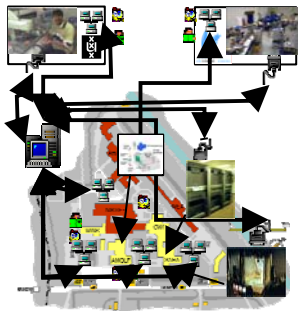


↓

Grids and P2P

↓

Virtual laboratories

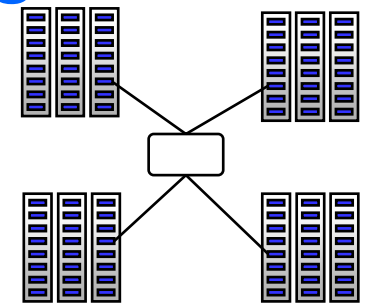


Examples cluster computing

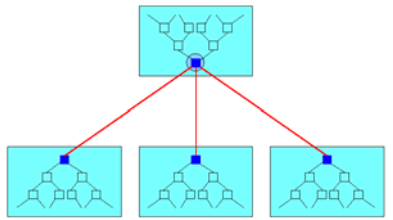
- **Communication protocols for Myrinet**
- **Parallel languages (Orca, Spar)**
- **Parallel applications**
  - PILE: Parallel image processing
  - HIRLAM: Weather forecasting
  - Solving Awari (3500-year old game)
- **GRAPE: N-body simulation hardware**

Distributed supercomputing on DAS

- **Study non-trivially parallel applications**
- **Grids usually are hierarchical**
  - Collections of clusters, supercomputers
  - Fast local links, slow wide-area links

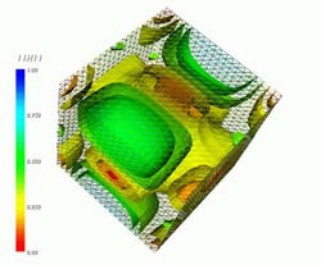


- **Can optimize algorithms to exploit hierarchy**
  - Message combining + latency hiding on wide-area links
  - Collective operations for wide-area systems
  - Load balancing



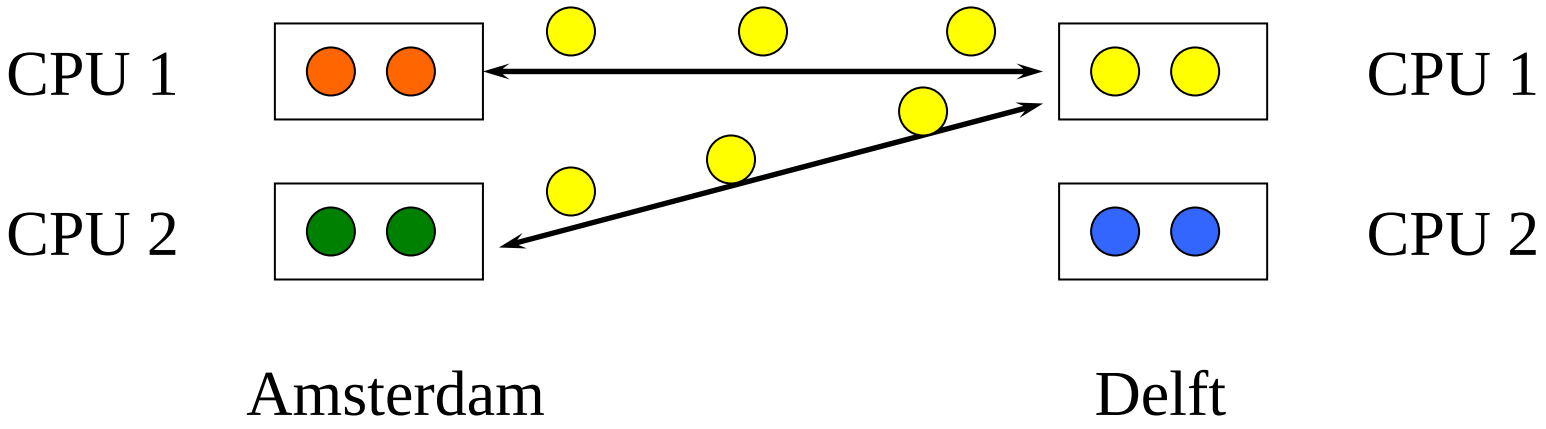
- **Did many successful experiments**

  [HPCA 1999, IEEE TPDS 2002, SC'04]

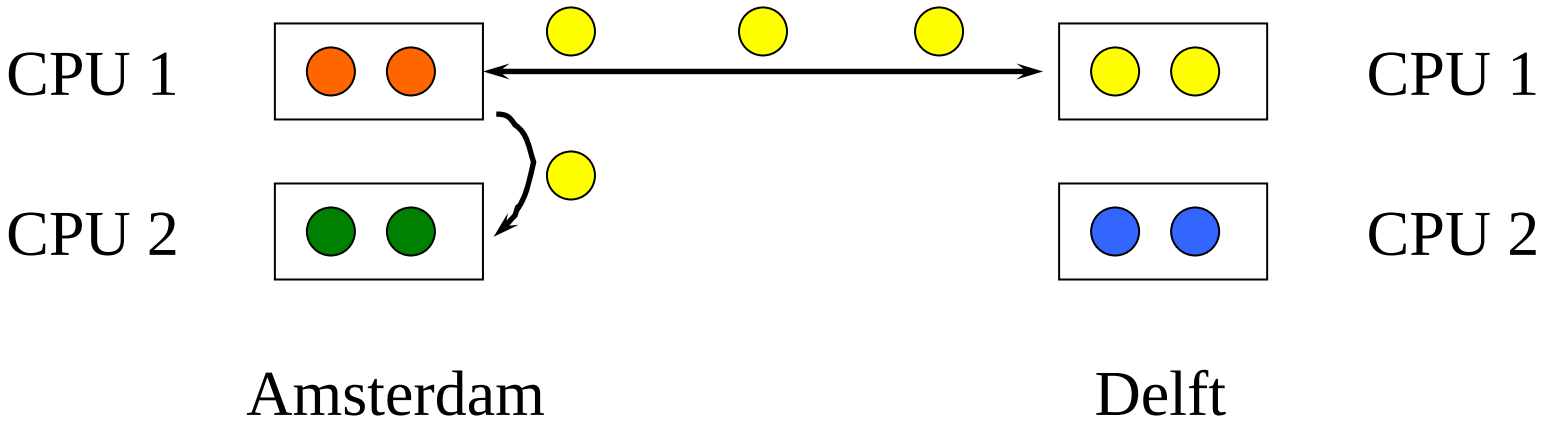


Example: N-body simulation

- **Much wide-area communication**
  - Each node needs info about remote bodies

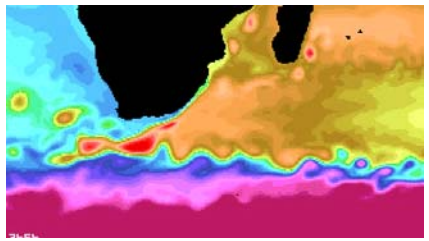


Trivial optimization



Example projects

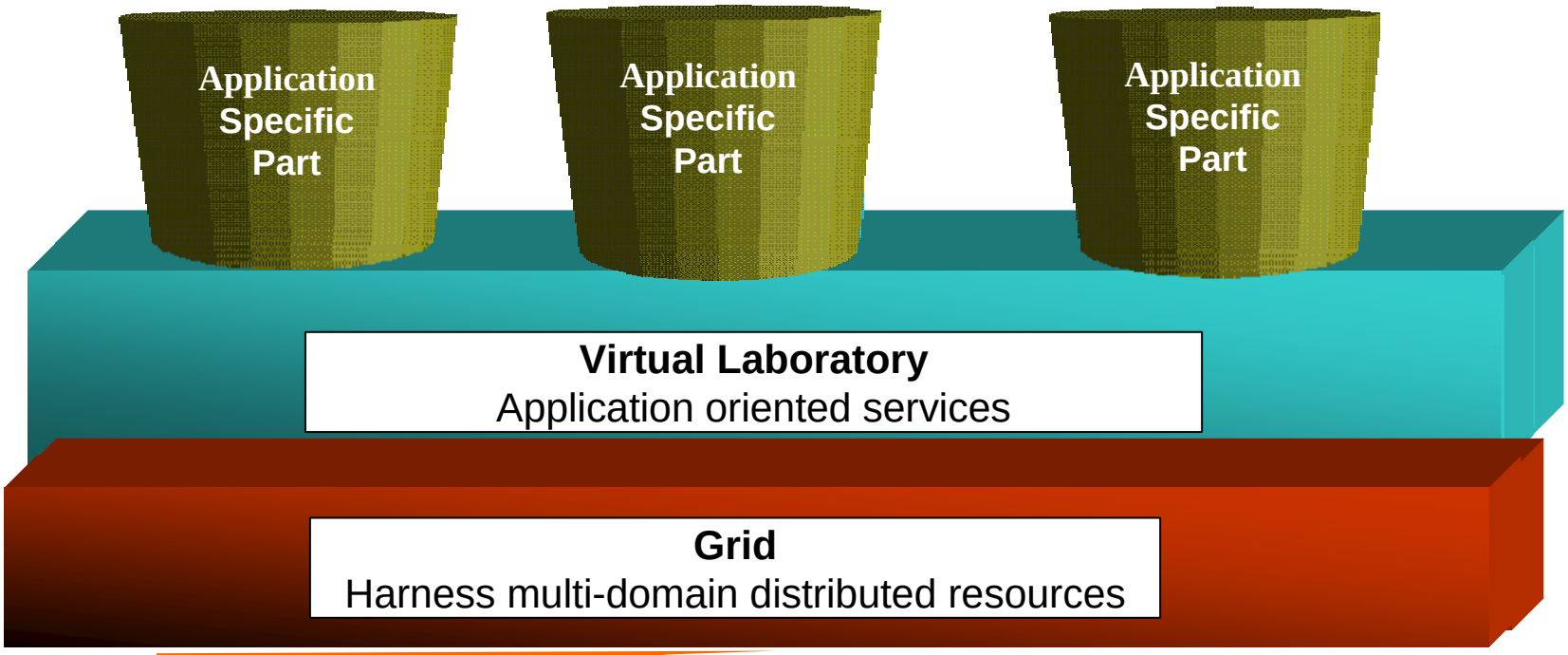
- **Albatross**
  - Optimize algorithms for wide area execution
- **MagPIe:**
  - MPI collective communication for WANs
- **Manta: distributed supercomputing in Java**
- **Dynamite: MPI checkpointing & migration**
- **ProActive (INRIA)**
- **Co-allocation/scheduling in multi-clusters**
- **Ensflow**
  - Stochastic ocean flow model

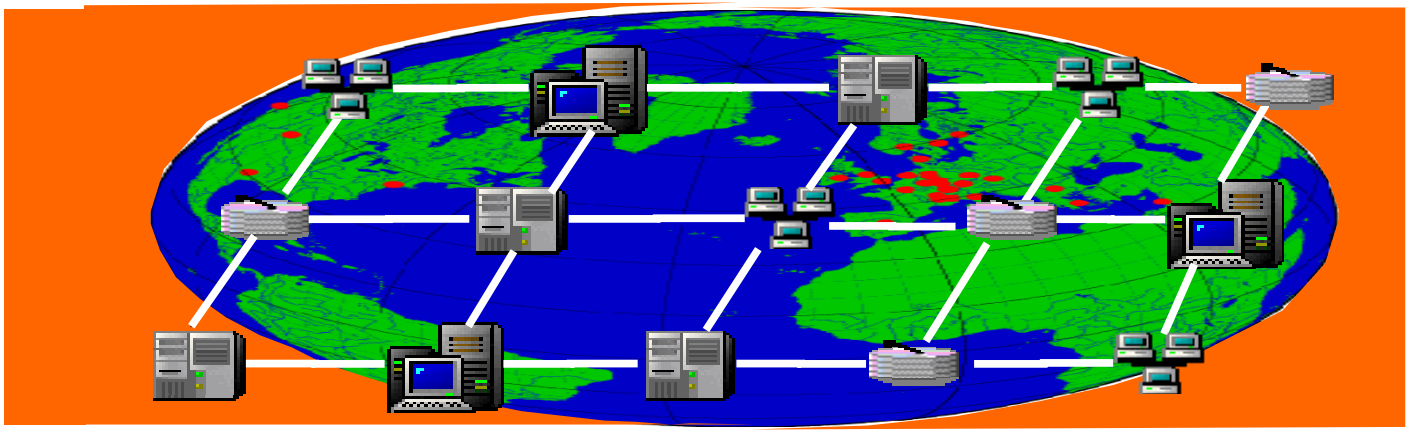


Grid & P2P computing: using DAS-2 as part of larger heterogeneous grids

- Ibis: Java-centric grid computing
- Satin: divide-and-conquer on grids
- Zorilla: P2P distributed supercomputing
- KOALA: co-allocation of grid resources
- Globule: P2P system with adaptive replication
- CrossGrid: interactive simulation and visualization of a biomedical system

Virtual Laboratories





VL-e: Virtual Laboratory for e-Science project (2004-2008)

- **40 M€ Dutch project (20 M€ from government)**
- **2 experimental environments:**
  - Proof of Concept: applications research
  - Rapid Prototyping (using DAS): computer science
- **Research on:**
  - Applications (biodiversity, bioinformatics, food informatics, telescience, physics)
  - Computer science tools for visualization, workflow, ontologies, data management, PSEs, grid computing

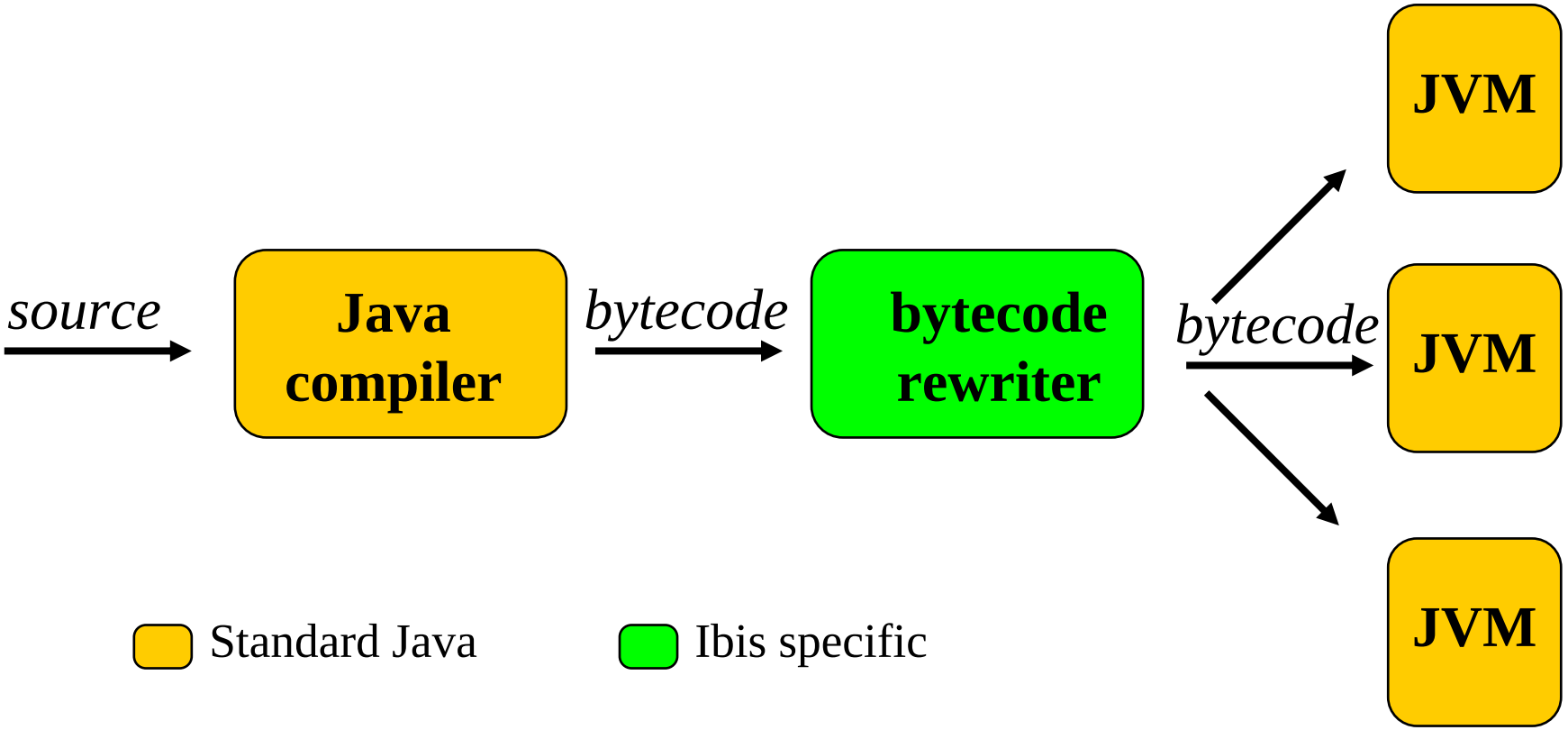
Outline

- History
- Impact on Dutch computer science research
  - Trend: cluster computing → distributed computing → Grids → Virtual laboratories
- **Example research projects**
  - Ibis, Satin
- **Grid experiments on DAS-2, GridLab, Grid'5000**
- **Future:** DAS-3

The Ibis system

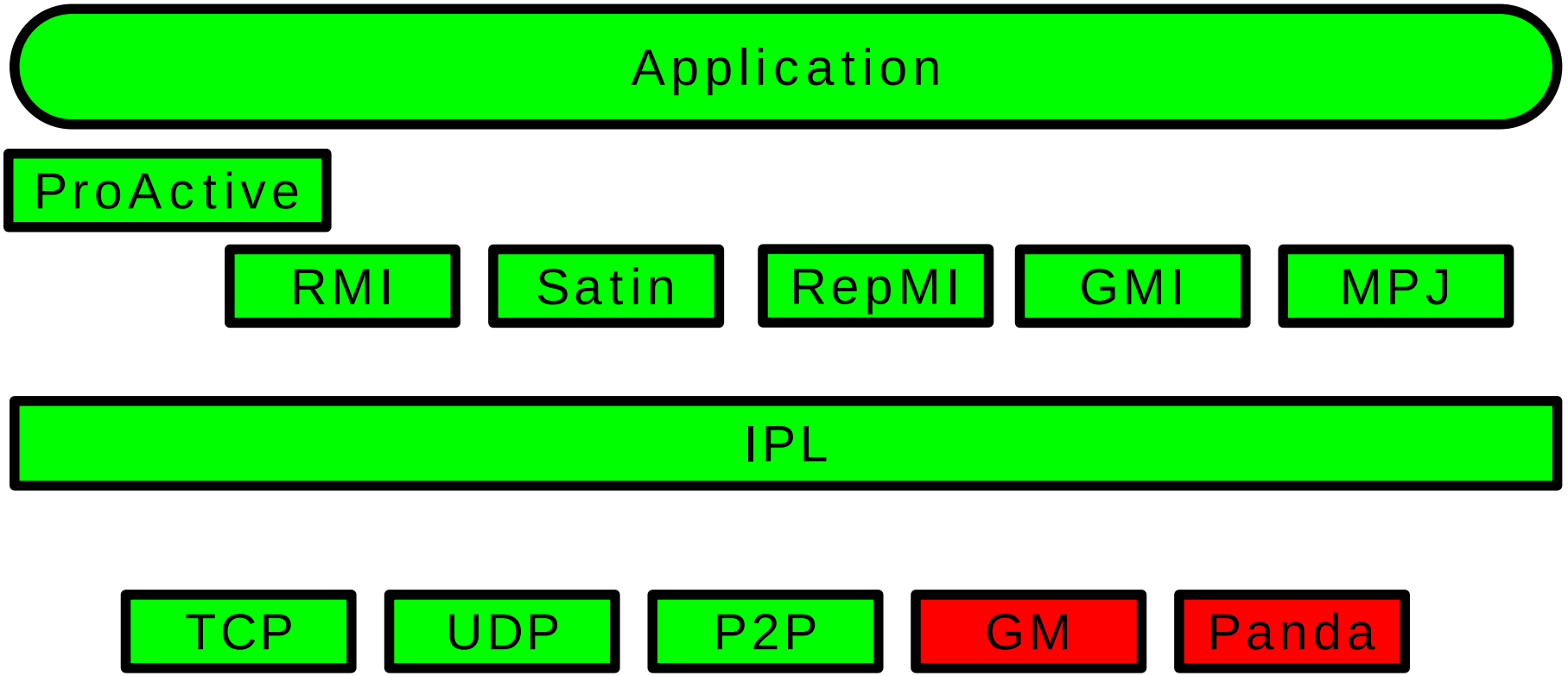
- **Programming support for distributed supercomputing on heterogeneous grids**
  - Fast RMI, group communication, object replication, d&c
- **Use Java-centric approach + JVM technology**
  - Inherently more portable than native compilation
  - Requires entire system to be written in pure Java
  - Optimized special-case solutions with native code
- **Several external users:**
  - ProActive, VU medical center, AMOLF, TU Darmstadt

Compiling/optimizing programs



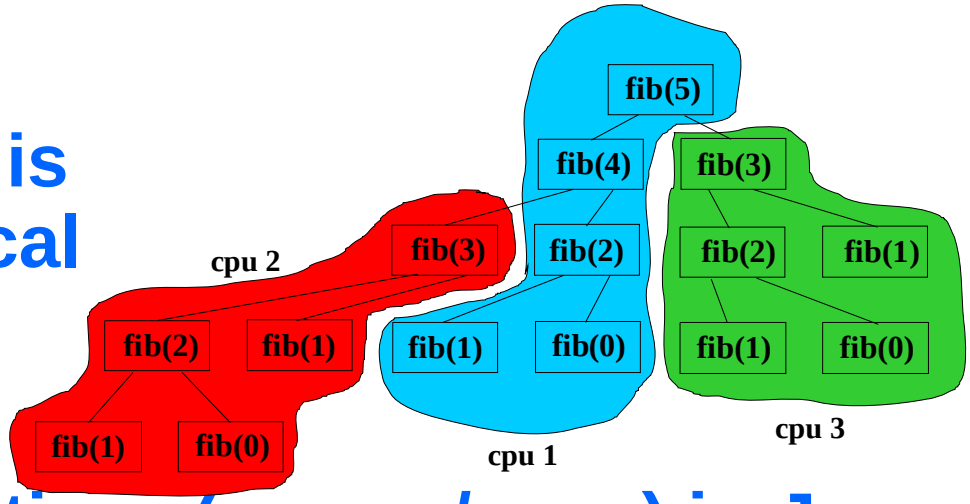
- **Optimizations are done by bytecode rewriting**
  - E.g. compiler-generated serialization

Ibis Overview



Satin: a parallel divide-and-conquer system on top of Ibis

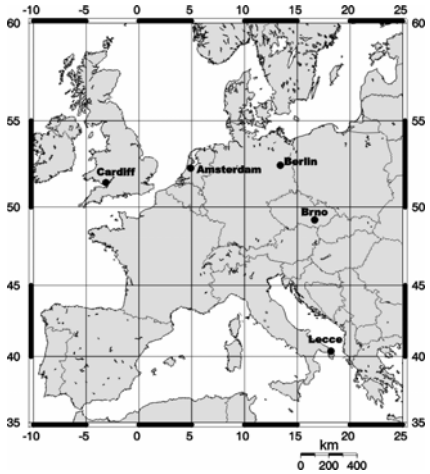
- Divide-and-conquer is inherently hierarchical
- More general than master/worker

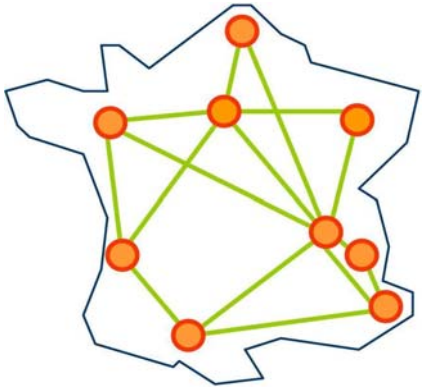


- Satin: Cilk-like primitives (spawn/sync) in Java
- Supports replicated shared objects with user-defined coherence semantics
- Supports malleability (nodes joining/leaving) and fault-tolerance (nodes crashing)

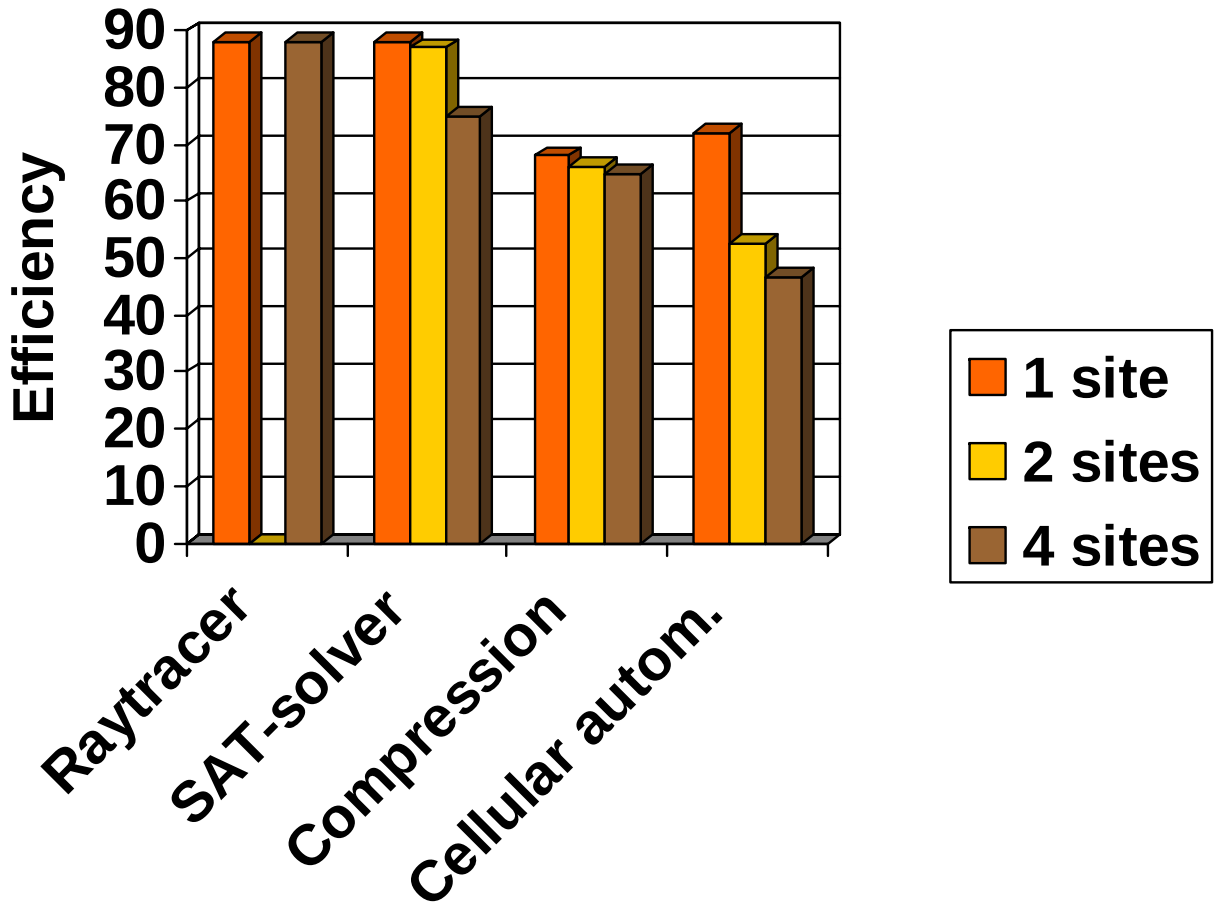
Grid experiments

- DAS is ``ideal'', laboratory-like environment for doing clean performance experiments
- GridLab testbed (incl. VU-DAS) is used for doing heterogeneous experiments
- Grid'5000 is used for large-scale experiments





Performance Ibis on wide-area DAS-2 (64 nodes)



- Cellular Automaton uses Ibis/IPL, the others use Satin.

GridLab

GridLab testbed

Testbed sites

Type	OS	CPU	Location	CPUs
Cluster	Linux	Pentium-3	Amsterdam	8 × 1
SMP	Solaris	Sparc	Amsterdam	1 × 2
Cluster	Linux	Xeon	Brno	4 × 2
SMP	Linux	Pentium-3	Cardiff	1 × 2
Origin 3000	Irix	MIPS	ZIB Berlin	1 × 16
Cluster	Linux	Xeon	ZIB Berlin	1 x 2
SMP	Unix	Alpha	Lecce	1 × 4
Cluster	Linux	Itanium	Poznan	1 x 4
Cluster	Linux	Xeon	New Orleans	2 x 2

Experiences

- **Grid testbeds are difficult to obtain**
- **Poor support for co-allocation**
- **Firewall problems everywhere**
- **Java indeed runs anywhere**
- **Divide-and-conquer parallelism can obtain high efficiencies (66-81%) on a grid**
  - See [van Reeuwijk, Euro-Par 2005]

GridLab results

Program	sites	CPUs	Efficiency
Raytracer (Satin)	**5**	**40**	**81 %**
SAT-solver (Satin)	**5**	**28**	**88 %**
Compression (Satin)	**3**	**22**	**67 %**
Cellular autom. (IPL)	**3**	**22**	**66 %**

- **Efficiency normalized to single CPU type (1GHz P3)**

Grid'5000 experiments

- **Used Grid'5000 for**
  - Nqueens challenge (2nd Grid Plugtest)
  - Testing Satin's shared objects
  - Large-scale P2P (Zorilla) experiments
- **Issues**
  - No DNS-resolution for compute nodes
  - Using local IP addresses (192.168.x.y) for routing
  - Setting up connections to nodes with multiple IP addresses
  - Unable to run on Grid'5000 and DAS-2 simultaneously

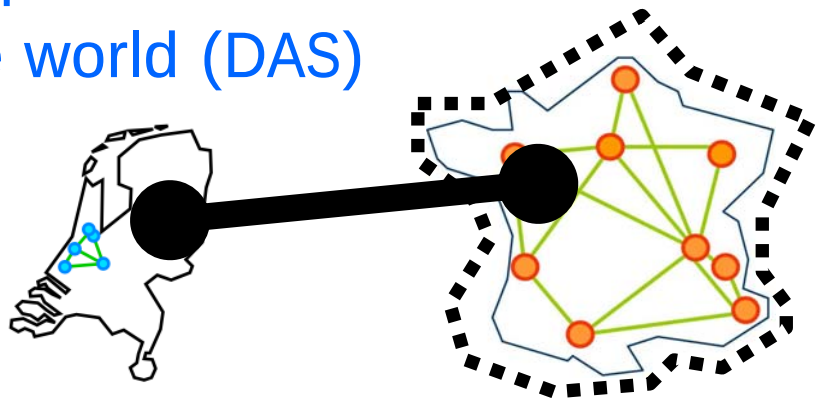
Grid'5000 results

Program	sites	CPUs	Efficiency
SAT solver	**3**	**112**	**56 %**
Traveling Salesman	**3**	**120**	**86 %**
VLSI routing	**3**	**120**	**84 %**
N-queens	**5**	**960**	**(~ 85 %)**

- **Satin programs (using shared objects)**
- **Running concurrently on clusters at Sophia-Antipolis, Bordeaux, Rennes, and Orsay**

Comparison

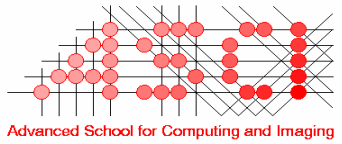
- **DAS**
  - + Ideal for speedup measurements (homogeneous)
  - - Bad for heterogeneous or long-latency experiments
- **GridLab testbed**
  - + Useful for heterogeneous experiments (e.g. Ibis)
  - - Small-scale, unstable
- **Grid'5000**
  - + Excellent for large-scale experiments
  - - Bad connectivity to outside world (DAS)



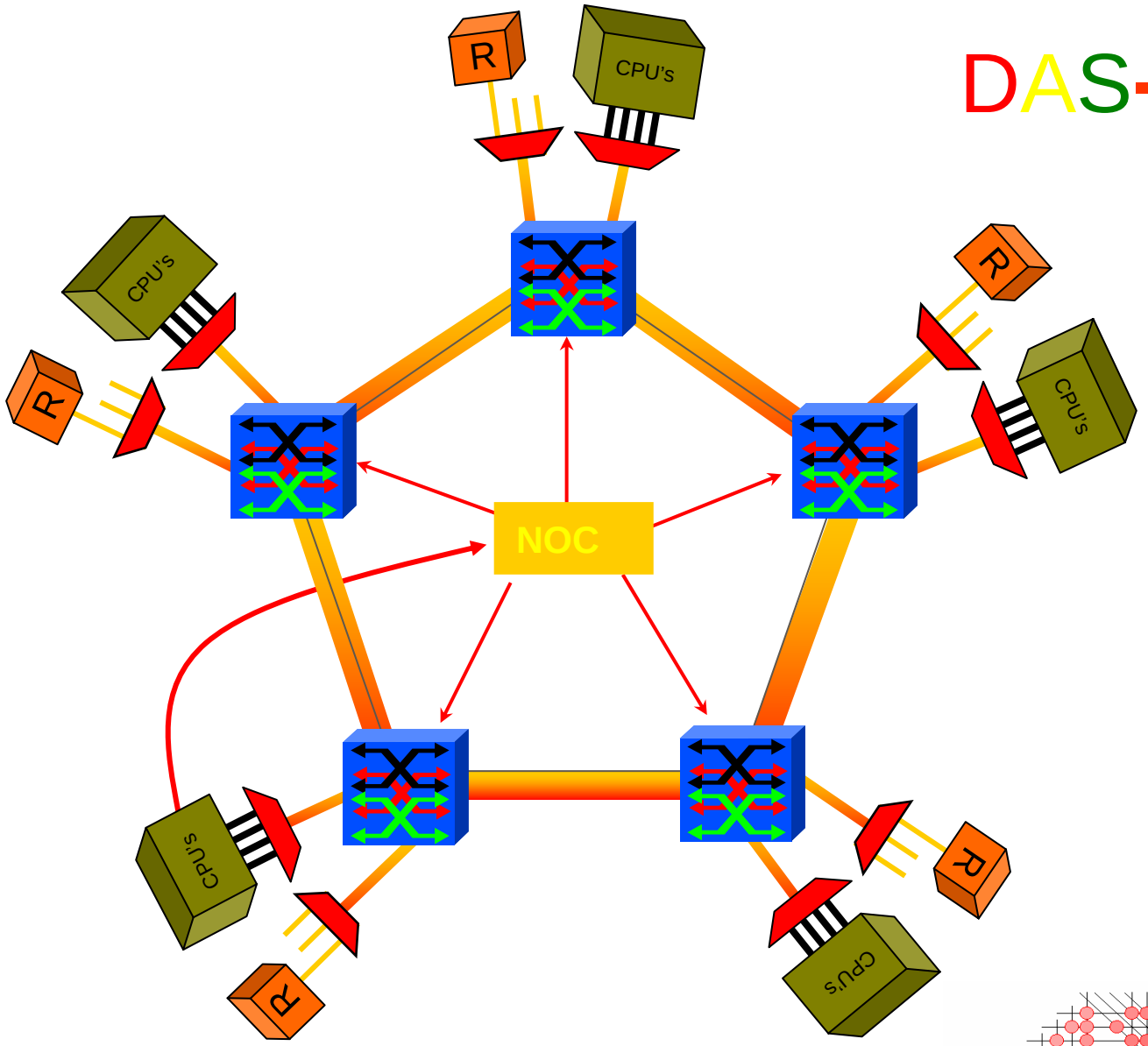
DAS-3 (2006)

- **Partners:**
  - ASCI, Gigaport-NG/SURFnet, VL-e, MultimediaN
- **Expected to be more heterogeneous**
- **Experiment with (nightly) production use**
- **DWDM backplane**
  - Dedicated optical group of lambdas
  - Can allocate multiple 10 Gbit/s lambdas between sites

GigaPort



DAS-3



vl·e

GigaPort

Advanced School for Computing and Imaging

StarPlane project

- **Key idea:**
  - Applications can dynamically allocate light paths
  - Applications can change the topology of the wide-area network, possibly even at sub-second timescale
- **Challenge: how to integrate such a network infrastructure with (e-Science) applications?**

- **(Collaboration with Cees de Laat, Univ. of Amsterdam)**

vl·e

GigaPort

Advanced School for Computing and Imaging

Conclusions

- **DAS is a shared infrastructure for experimental computer science research**
- **It allows controlled (laboratory-like) grid experiments**
- **It accelerated the research trend**
  - cluster computing ➔ distributed computing ➔ Grids ➔ Virtual laboratories
- **We want to use DAS as part of larger international grid experiments (e.g. with Grid'5000)**

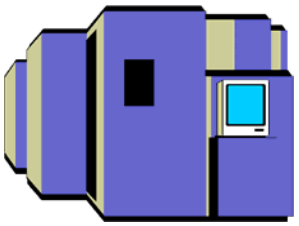
Acknowledgements

- Grid'5000
- Andy Tanenbaum
- Bob Hertzberger
- Henk Sips
- Lex Wolters
- Dick Epema
- Cees de Laat
- Aad van der Steen
- Rob van Nieuwpoort
- Jason Maassen
- Kees Verstoep
- Gosia Wrzesinska
- Niels Drost
- Thilo Kielmann
- Ceriel Jacobs
- Many others

More info: http://www.cs.vu.nl/das2/

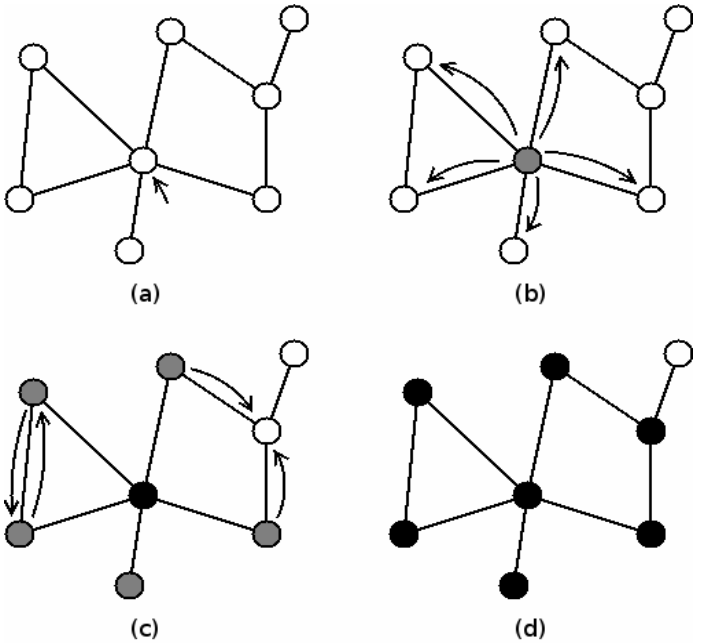
Funding

- **Funded mainly by NWO (Dutch national science foundation)**
- **Motivation: CS needs its own infrastructure for**
  - Systems research and experimentation
  - Distributed experiments
  - Doing many small, interactive experiments
- **Need distributed experimental system, rather than centralized production supercomputer**



Zorilla: P2P supercomputing

- Fully distributed Java-based system for running parallel applications
- Uses P2P techniques
- Supports malleable (Satin) applications
- Uses locality-aware flood-scheduling algorithm



Running applic's *without* Zorilla

- **Deployment**
  - Copy program and input files to all sites
  - Determine local job scheduling mechanism, write job submission scripts
  - Determine network setup of all clusters
- **Running**
  - Determine site and node availability
  - Submit application to the scheduler on each site
  - Monitor progress of application
- **Clean up**
  - Gather output and log files
  - Cancel remaining reservations
  - Remove program, input, output and log files from sites

Running applic's *with* Zorilla

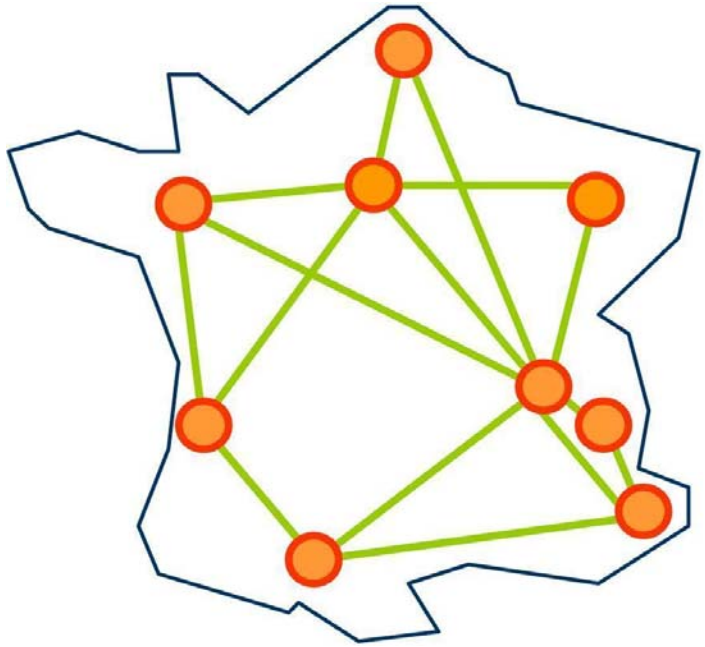
- **Deployment (once)**
  - Copy Zorilla to all sites
  - Determine local job scheduling mechanism, write job submission scripts
  - Determine network setup of all clusters
  - Submit Zorilla to local job schedulers
- **Running and Clean up**
  - Submit job to Zorilla system

```
$ zubmit -j nqueens.jar -#w676 NQueens 1 22 5
```

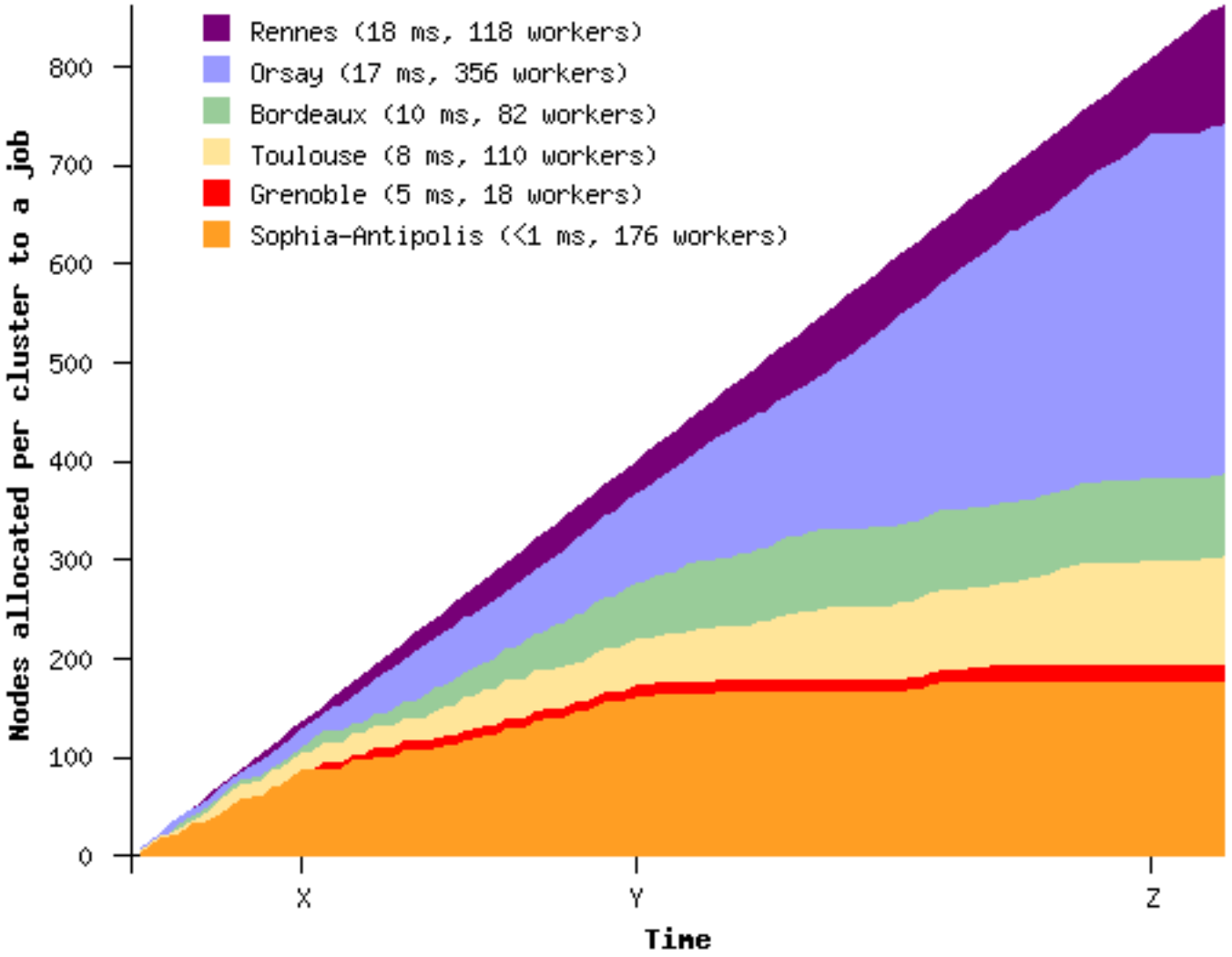
Zorilla on Grid’5000

- N-Queens divide-and-conquer Java application
- Six Grid’5000 clusters
- 338 machines
- 676 processors

- 22-queens in 35 minutes



Processor allocation results



Education using DAS

- **International (top)master program PDCS: Parallel and Distributed Computer Systems**
- **Organized by Andy Tanenbaum, Henri Bal, Maarten van Steen et al.**
- **See http://www.cs.vu.nl/masters/compsys/**

vrije Universiteit