DAS 1-4:
Experiences with the Distributed ASCI Supercomputers

Henri Bal
bal@cs.vu.nl

Kees Verstoep
versto@cs.vu.nl

Vrije Universiteit Amsterdam

vrije Universiteit

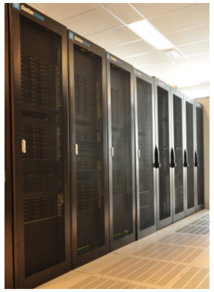
Introduction

- DAS: shared distributed infrastructure for experimental computer science research
  - Controlled experiments for CS, *not production*
- 16 years experience with funding & research
- Huge impact on Dutch CS

DAS-1

DAS-2

DAS-3



DAS-4

Overview

- Historical overview of the DAS systems
- How to organize a national CS testbed
- Examples of research done on DAS at the VU
  - Ask Adam Belloum for an UvA perspective ☺

vrije Universiteit

Outline

- DAS (pre-)history
- DAS organization
- DAS-1 – DAS-4 systems
- DAS-1 research
- DAS-2 research
- DAS-3 research
- DAS-4 research
- DAS conclusions

Historical perspective

Mainframes
Minicomputers

Metacomputing paper
Beowulf cluster
Flocking Condors
Grid blueprint book

Clouds
Accelerators
Green IT
Big Data

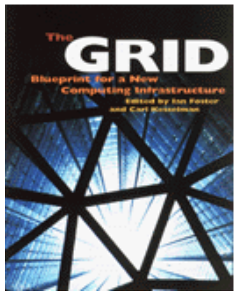
1980 — 1990 — 2000 — 2010

Pre-DAS

DAS

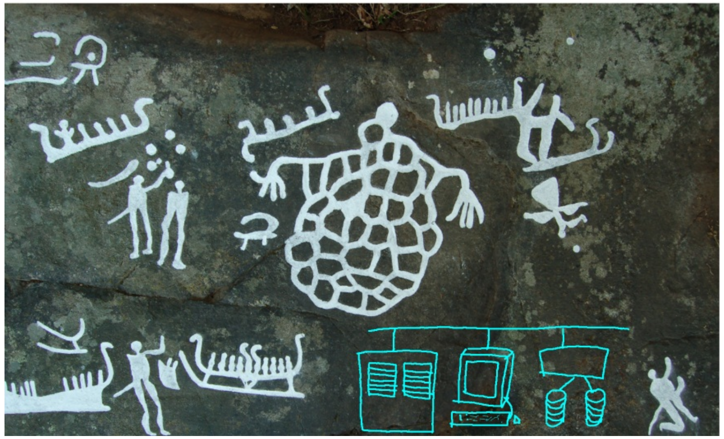
Collections of workstations (COWs)
Networks of workstations (NOWs)
Processor pools, ➔***Clusters***

Grids
e-Science
Optical networks



VU (pre-)history

- Andy Tanenbaum already built a cluster around 1984
  - Pronet token ring network
  - 8086 CPUs
  - (no pictures available)



- He built several Amoeba processor pools
  - MC68000, MC68020, MicroSparc
  - VME bus, 10 Mb/s Ethernet, Myrinet, ATM

Amoeba processor pool (Zoo, 1994)

vrije Universiteit

DAS-1 background: ASCI

- Research schools (Dutch product from 1990s)
  - Stimulate top research & collaboration
  - Organize Ph.D. education
- ASCI (1995):
  - Advanced School for Computing and Imaging
  - About 100 staff & 100 Ph.D. students from TU Delft, Vrije Universiteit, U. of Amsterdam, Leiden, Utrecht, TU Eindhoven, TU Twente, …

Motivation for DAS-1

- CS/ASCI 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

vrije Universiteit

Funding

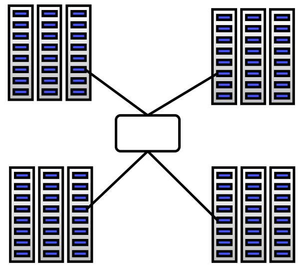
- DAS proposals written by ASCI committees
  - Chaired by Tanenbaum (DAS-1), Bal (DAS 2-5)
- NWO (national science foundation) funding for all 4 DAS systems (100% success rate)
  - About 900 K€ funding per system, 300 K€ matching by participants, extra funding from VU
  - ➔ ***Currently preparing for DAS-5***
- ASCI committee also acts as steering group

NWO
Netherlands Organisation for Scientific Research

vrije Universiteit

Goals of DAS-1

- Goals of DAS systems:
  - Ease collaboration within ASCI
  - Ease software exchange
  - Ease systems management
  - Ease experimentation

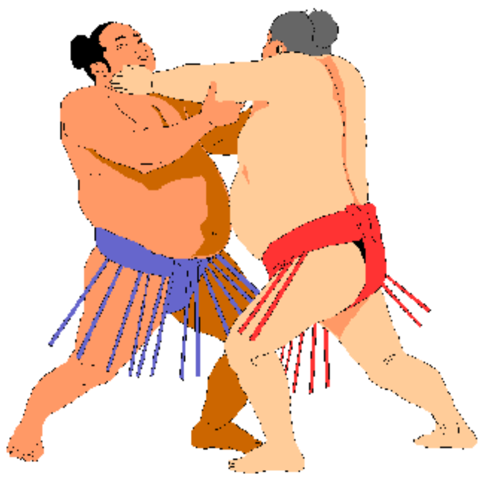


- Want a clean, laboratory-like system
- Keep DAS simple and *homogeneous*
  - Same OS, local network, CPU type everywhere
  - Single (replicated) user account file
- [ACM SIGOPS 2000] (paper with 50 authors)

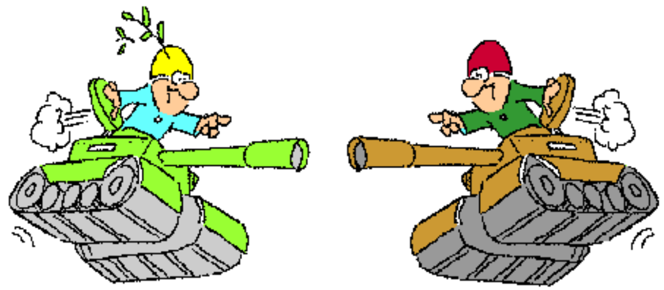
vrije Universiteit

Behind the screens ….

Artist's Rendition of the First OS Discussion



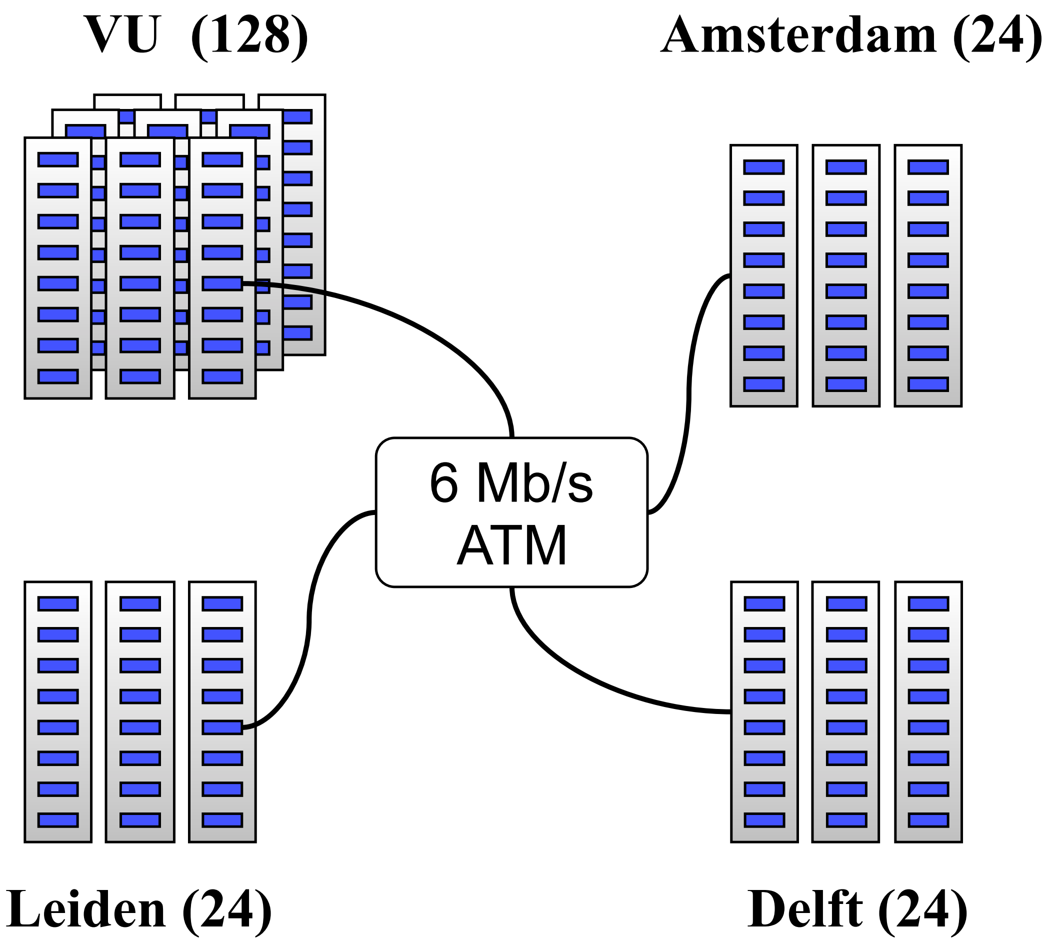
Artist's Rendition of the Second OS Discussion



Source: Tanenbaum (ASCI'97 conference)

DAS-1 (1997-2002)
A homogeneous wide-area system

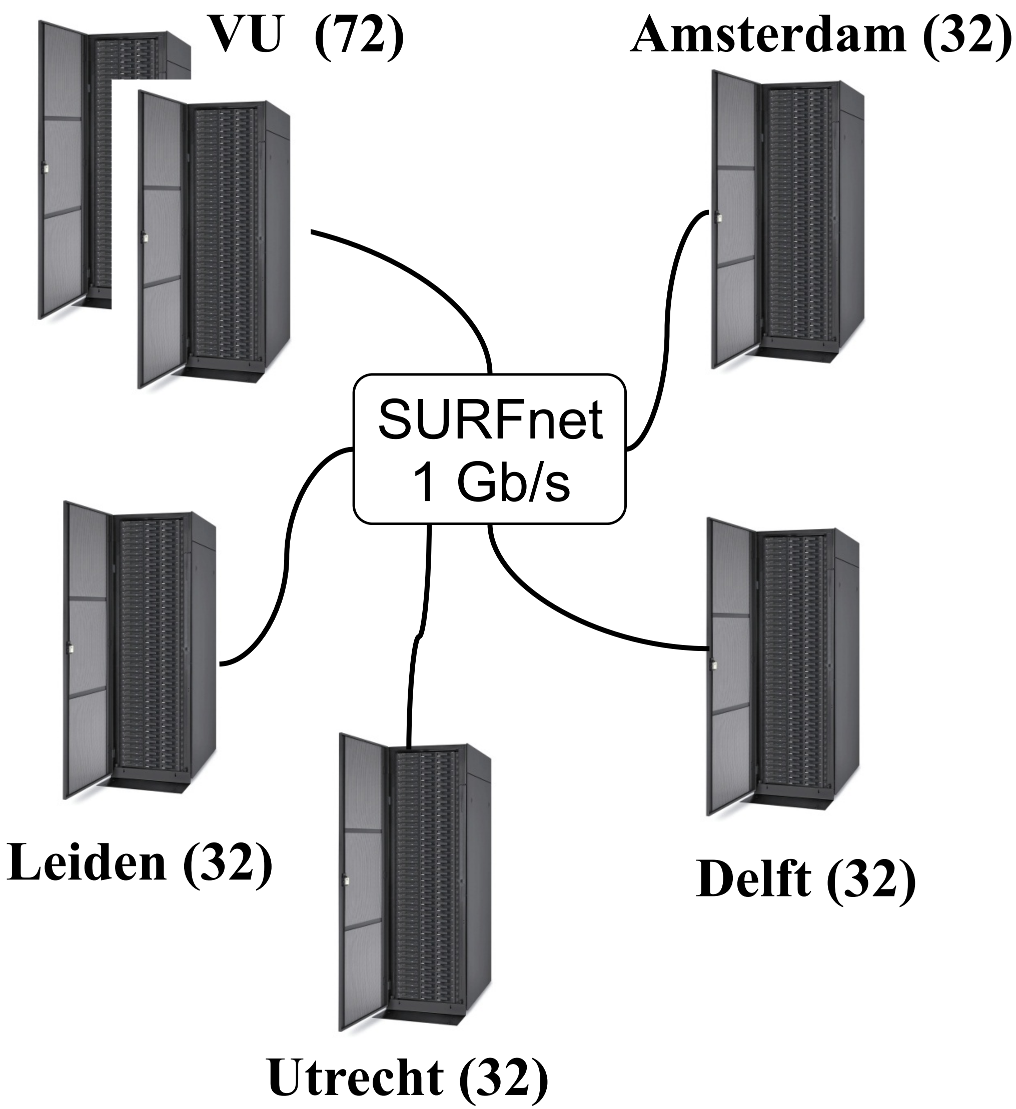
200 MHz Pentium Pro
128 MB memory
Myrinet interconnect
BSDI → RedHat Linux
Built by Parsytec



DAS-2 (2002-2007)
a Computer Science Grid

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

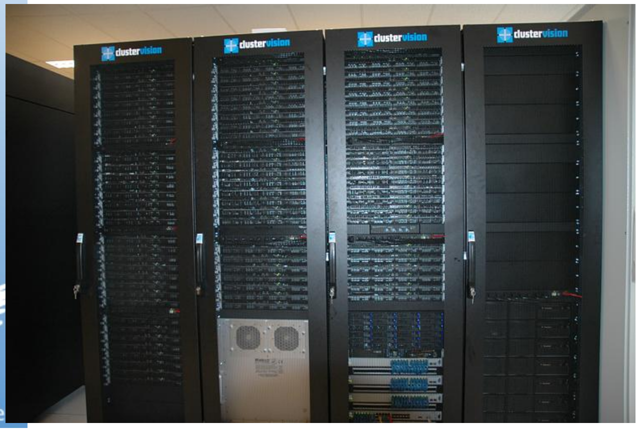
Myrinet interconnect
Redhat Enterprise Linux
Globus 3.2
PBS → Sun Grid Engine
Built by IBM

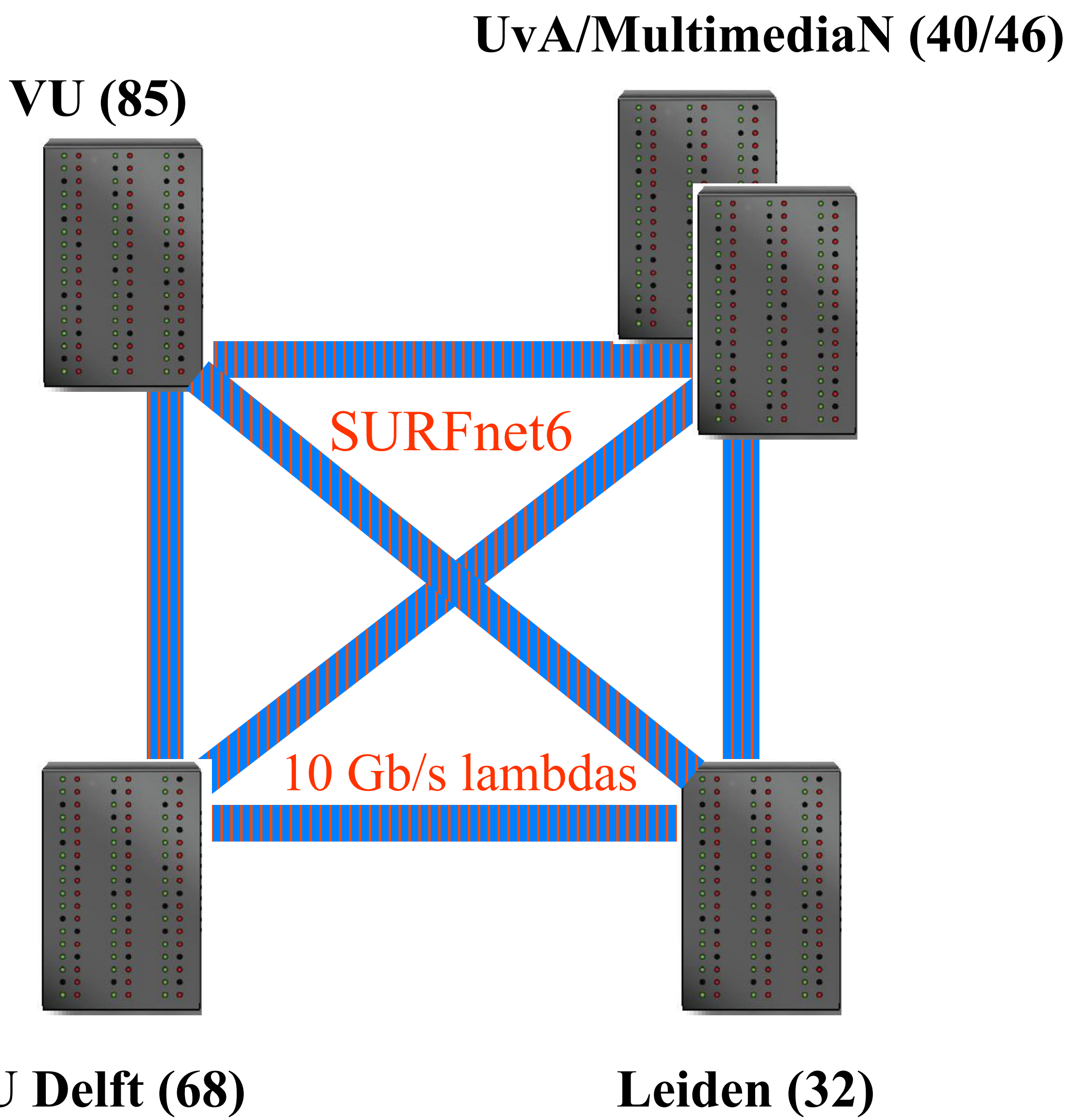


vrije Universiteit

DAS-3 (2007-2010)
An optical grid

- dual AMD Opterons
- 4 GB memory
- 250-1500 GB disk
- More heterogeneous:
  - 2.2-2.6 GHz
    - Single/dual core nodes
- Myrinet-10G (exc. Delft)
- Gigabit Ethernet
- Scientific Linux 4
- Globus, SGE
- Built by ClusterVision

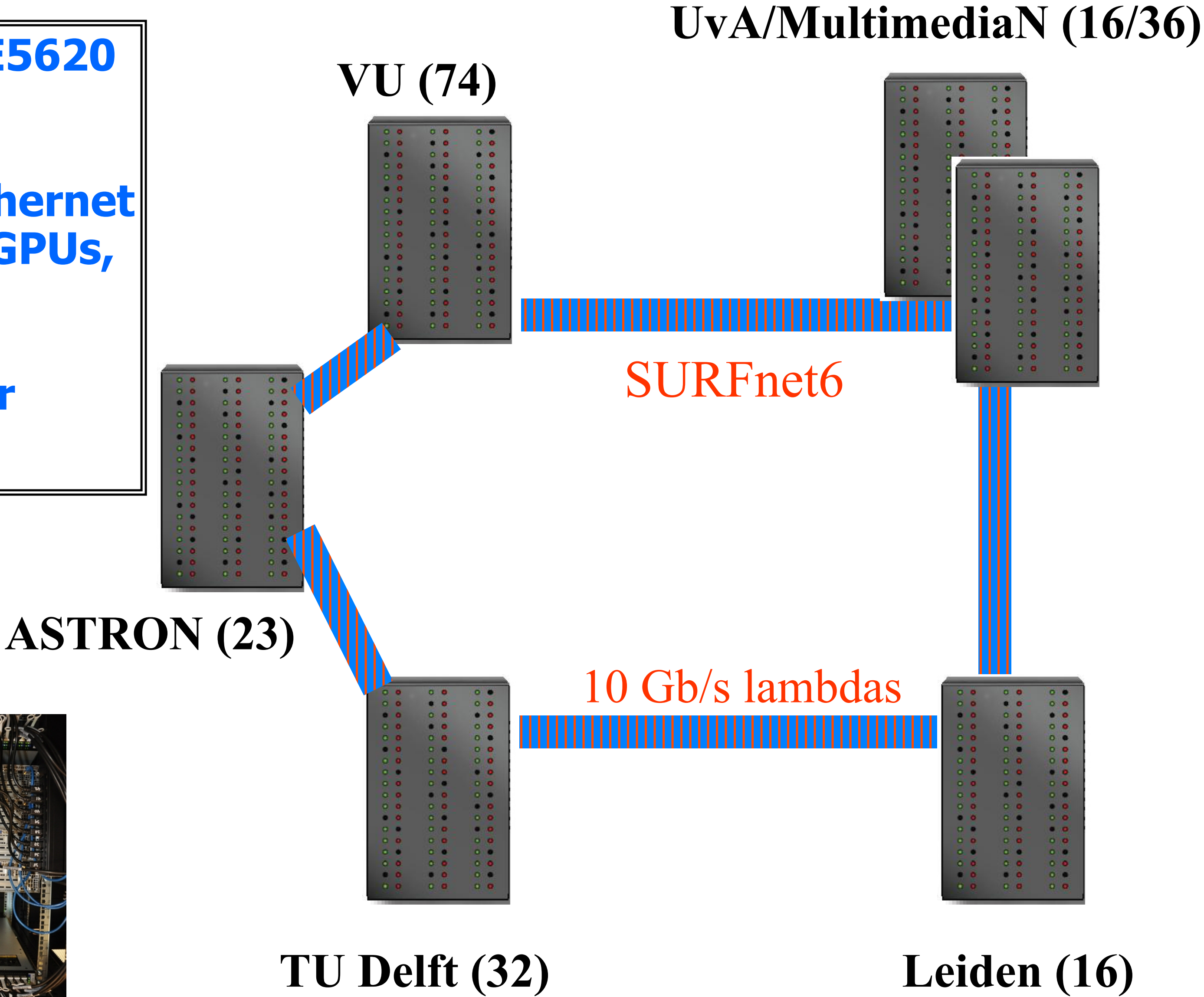


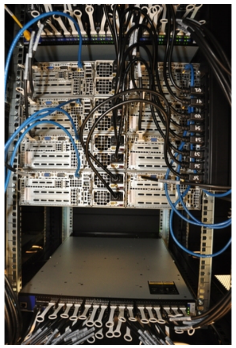


vrije Universiteit

DAS-4 (2011)
Testbed for clouds, diversity & Green IT

Dual quad-core Xeon E5620
24-48 GB memory
1-10 TB disk
Infiniband + 1Gb/s Ethernet
Various accelerators (GPUs, multicores, ...)
CentOS Linux 6
Bright Cluster Manager
Built by ClusterVision





vrije Universiteit

Performance

	DAS-1	DAS-2	DAS-3	DAS-4
# CPU cores	200	400	792	1600
SPEC CPU2000 INT (base)	78.5	454	1445	4620
SPEC CPU2000 FP (base)	69.0	329	1858	6160
1-way latency MPI (μs)	21.7	11.2	2.7	1.9
Max. throughput (MB/s)	75	160	950	2700
Wide-area bandwidth (Mb/s)	6	1000	40000	40000

vrije Universiteit

Impact of DAS

- Major incentive for VL-e ➔ 20 M€ funding
  - Virtual Laboratory for e-Science
- Collaboration SURFnet on DAS-3 & DAS-4
  - SURFnet provides dedicated 10 Gb/s light paths
- About 10 Ph.D. theses per year use DAS
- Many other research grants

Outline

- DAS (pre-)history
- DAS organization
- DAS-1 – DAS-4 systems
- DAS-1 research
- DAS-2 research
- DAS-3 research
- DAS-4 research
- DAS conclusions

vrije Universiteit

DAS research agenda

- Pre-DAS: Cluster computing
- DAS-1: Wide-area computing
- DAS-2: Grids & P2P computing
- DAS-3: e-Science & optical Grids
- DAS-4: Clouds, accelerators & green IT

vrije Universiteit

Overview VU research

	Algorithms & applications	*Programming systems*
DAS-1	Albatross	Manta MagPIe
DAS-2	Search algorithms Awari	Satin
DAS-3	StarPlane Model checking	Ibis
DAS-4	Multimedia analysis Semantic web	Ibis

vrije Universiteit

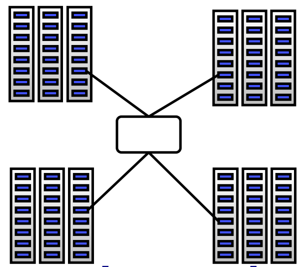
DAS-1 research

- Albatross: optimize wide-area algorithms
- Manta: fast parallel Java
- MagPIe: fast wide-area collective operations

vrije Universiteit

Albatross project

- Study algorithms and applications for wide-area parallel systems
- Basic assumption: wide-area system is hierarchical



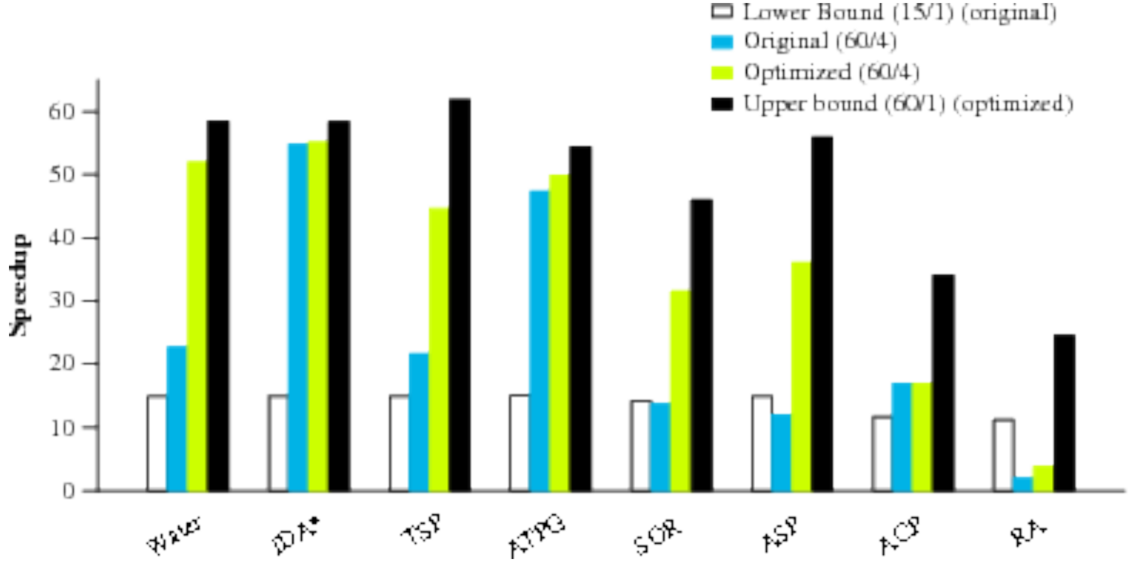
  - Connect clusters, not individual workstations
- General approach
  - Optimize applications to exploit hierarchical structure ➔ most communication is local

➢ [HPCA 1999]

vrije Universiteit

Wide-area algorithms

- Discovered numerous optimizations that reduce wide-area overhead
  - Caching, load balancing, message combining …
- Performance comparison between
  - 1 small (15 node) cluster, 1 big (60 node) cluster, wide-area (4*15 nodes) system



Manta: high-performance Java

- Native compilation (Java ➔ executable)
- Fast RMI protocol
  - Compiler-generated serialization routines
- Factor 35 lower latency than JDK RMI
- Used for writing wide-area applications

➢ [ACM TOPLAS 2001]

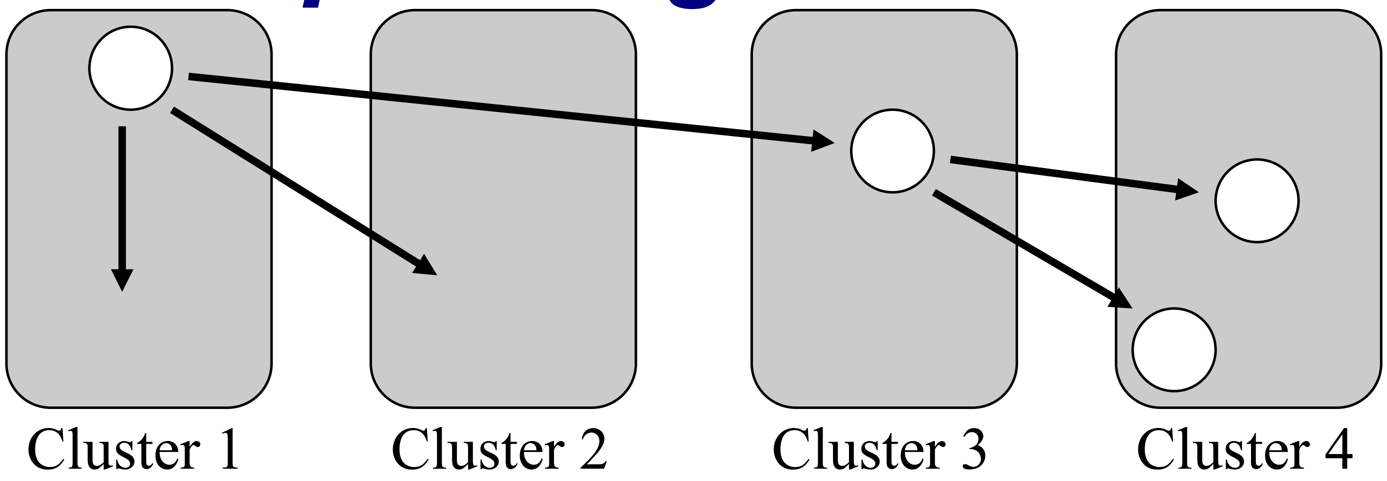
vrije Universiteit

MagPIe: wide-area collective communication

- Collective communication among many processors
  - e.g., multicast, all-to-all, scatter, gather, reduction
- MagPIe: MPI's collective operations optimized for hierarchical wide-area systems
- Transparent to application programmer

➢ [PPoPP'99]

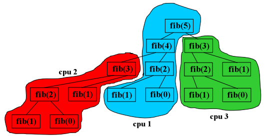
Spanning-tree broadcast



- MPICH (WAN-unaware)
  - Wide-area latency is chained
  - Data is sent multiple times over same WAN-link
- MagPIe (WAN-optimized)
  - Each sender-receiver path contains ≤1 WAN-link
  - No data item travels multiple times to same cluster

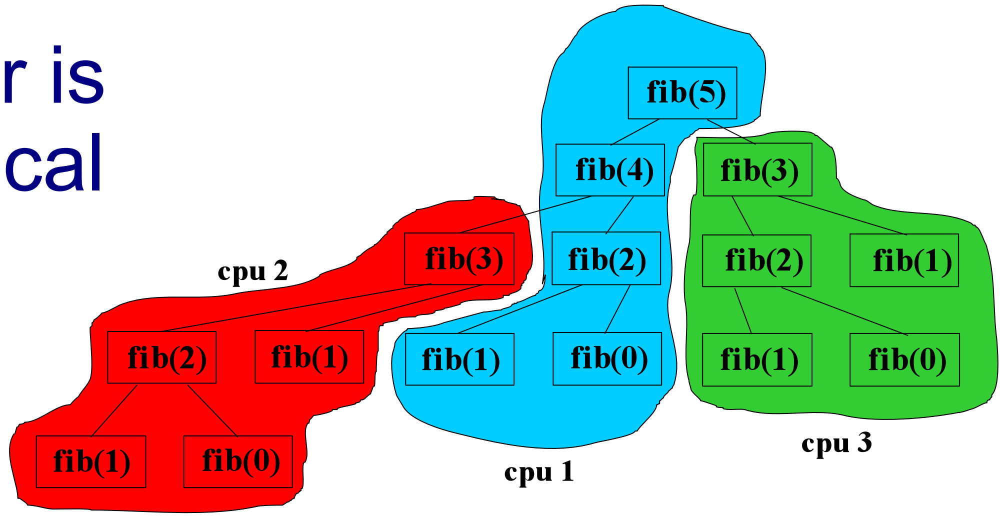
DAS-2 research

- Satin: wide-area divide-and-conquer
- Search algorithms
- Solving Awari



Satin: parallel divide-and-conquer

- Divide-and-conquer is inherently hierarchical
- More general than master/worker
- Satin: Cilk-like primitives (spawn/sync) in Java

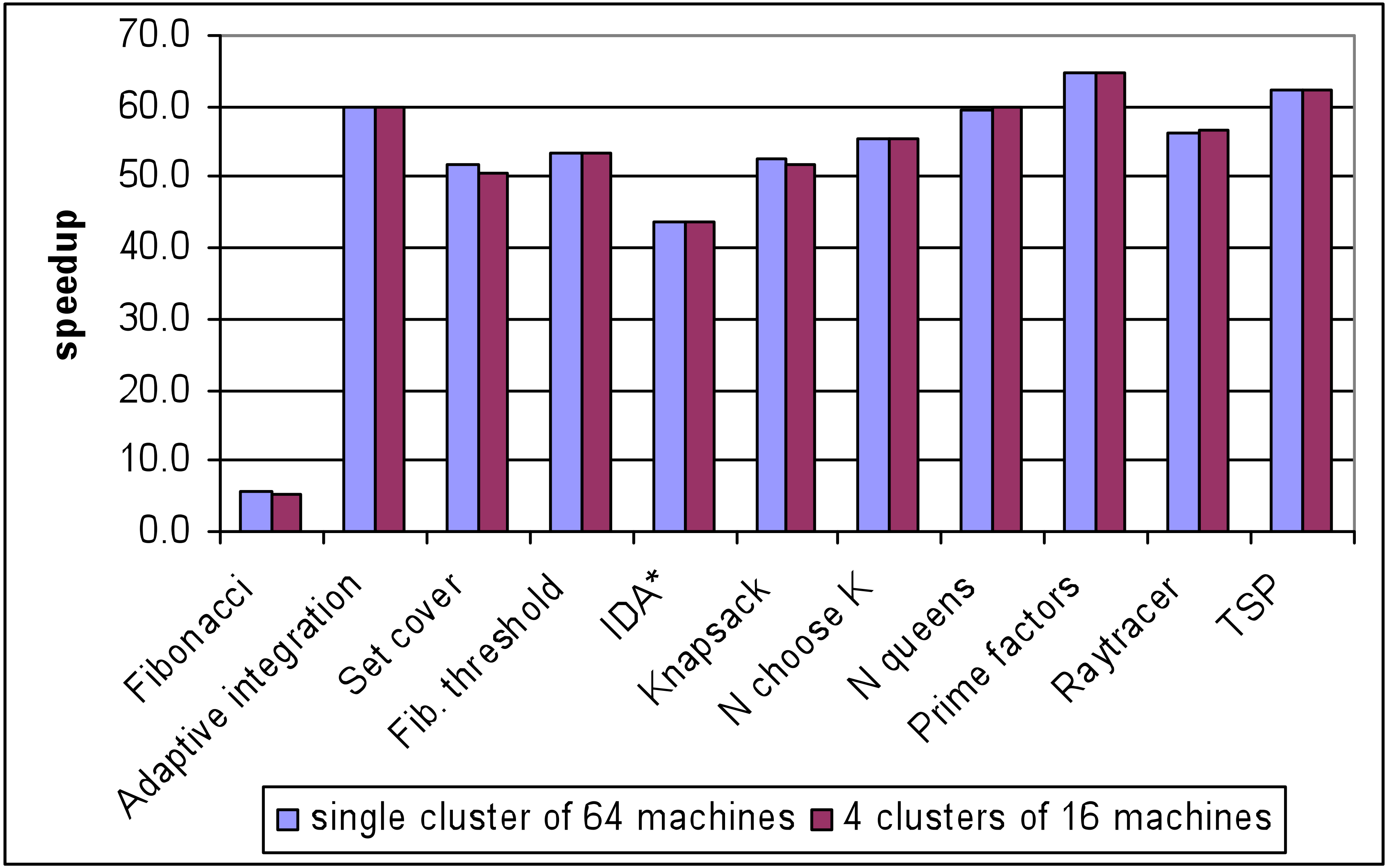


Satin

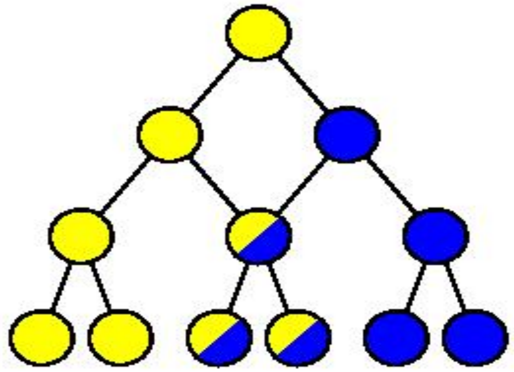
- Grid-aware load-balancing [PPoPP'01]
- Supports malleability (nodes joining/leaving) and fault-tolerance (nodes crashing) [IPDPS'05]
- Self-adaptive [PPoPP'07]
- Range of applications (SAT-solver, N-body simulation, raytracing, grammar learning, ….) [ACM TOPLAS 2010]

- Ph.D theses: van Nieuwpoort (2003), Wrzesinska (2007)

Satin on wide-area DAS-2



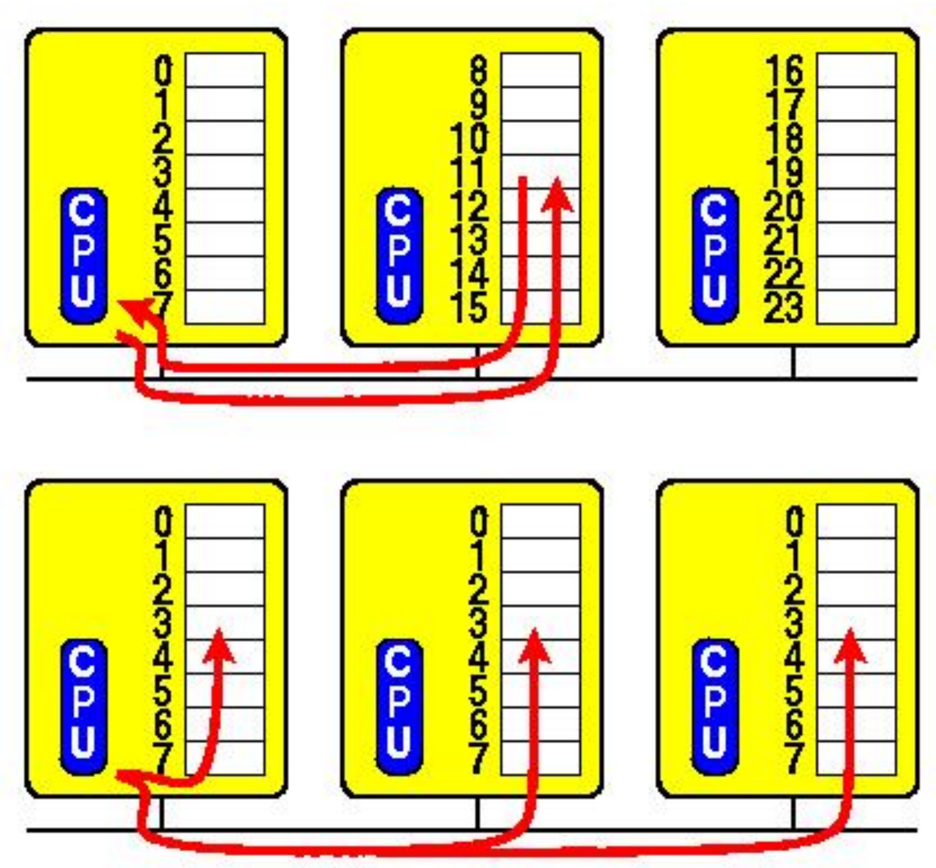
Search algorithms



- Parallel search:
  partition search space
- Many applications have *transpositions*
  - Reach same position through different moves
- Transposition table:
  - Hash table storing positions that have been evaluated before
  - Shared among all processors

vrije Universiteit

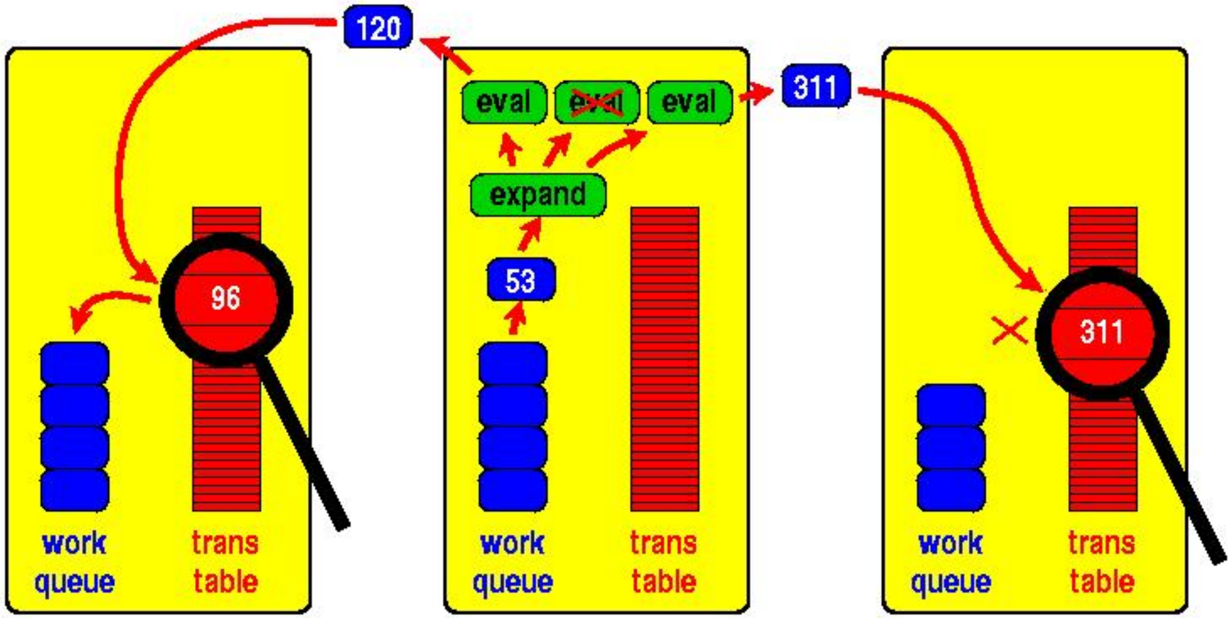
Distributed transposition tables



- Partitioned tables
  - 10,000s synchronous messages per second
  - Poor performance even on a cluster
- Replicated tables
  - Broadcasting doesn't scale (many updates)

vrije Universiteit

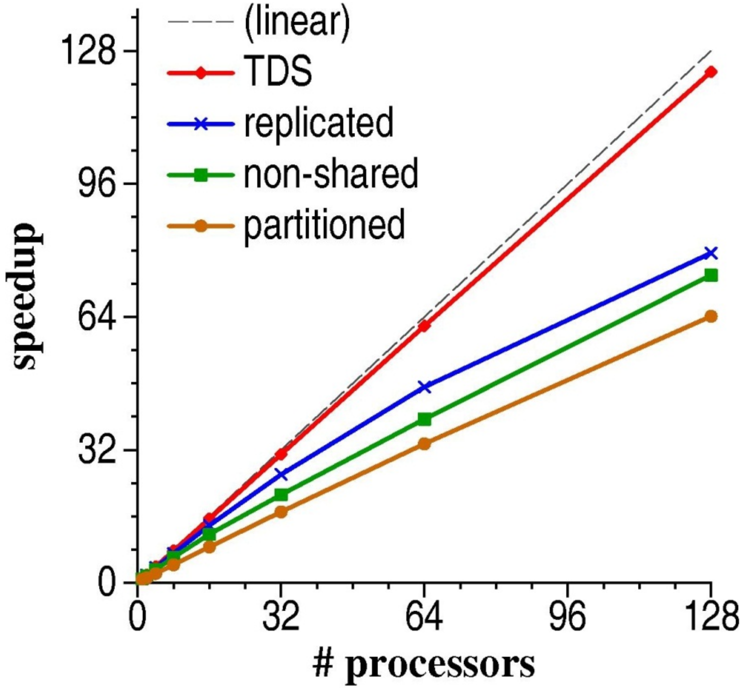
Transposition Driven Scheduling



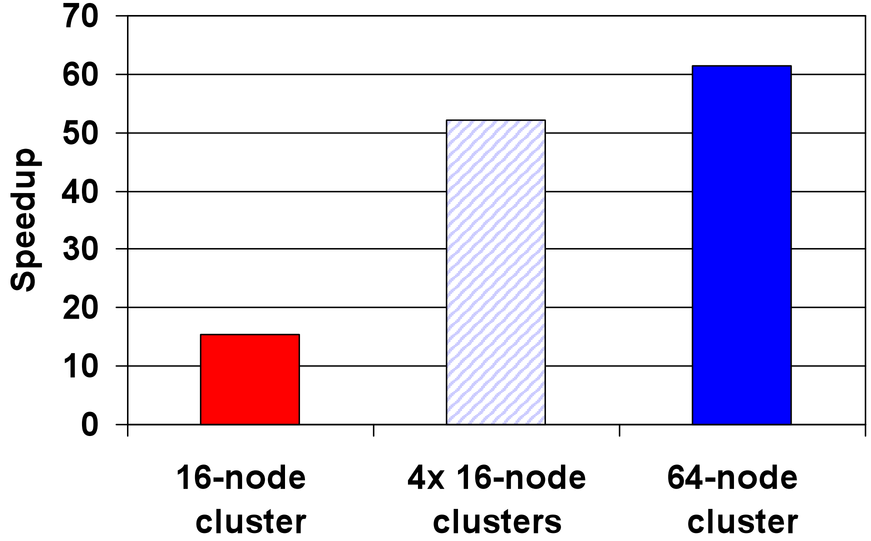
- Send job asynchronously to owner table entry
  - Can be overlapped with computation
  - Random (=good) load balancing
  - Delayed & combined into fewer large messages
    - ➔ Bulk transfers are far more efficient on most networks

Speedups for Rubik's cube

Single Myrinet cluster



TDS on wide-area DAS-2



- Latency-insensitive algorithm works well even on a grid, despite huge amount of communication
  - [IEEE TPDS 2002]

Solving awari

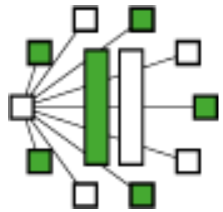
- Solved by John Romein [IEEE Computer, Oct. 2003]
  - Computed on VU DAS-2 cluster, using similar ideas as TDS
- Determined score for <u>889,063,398,406</u> positions
- Game is a draw

  Andy Tanenbaum:
  ``You just ruined a perfectly fine 3500 year old game''

vrije Universiteit

DAS-3 research

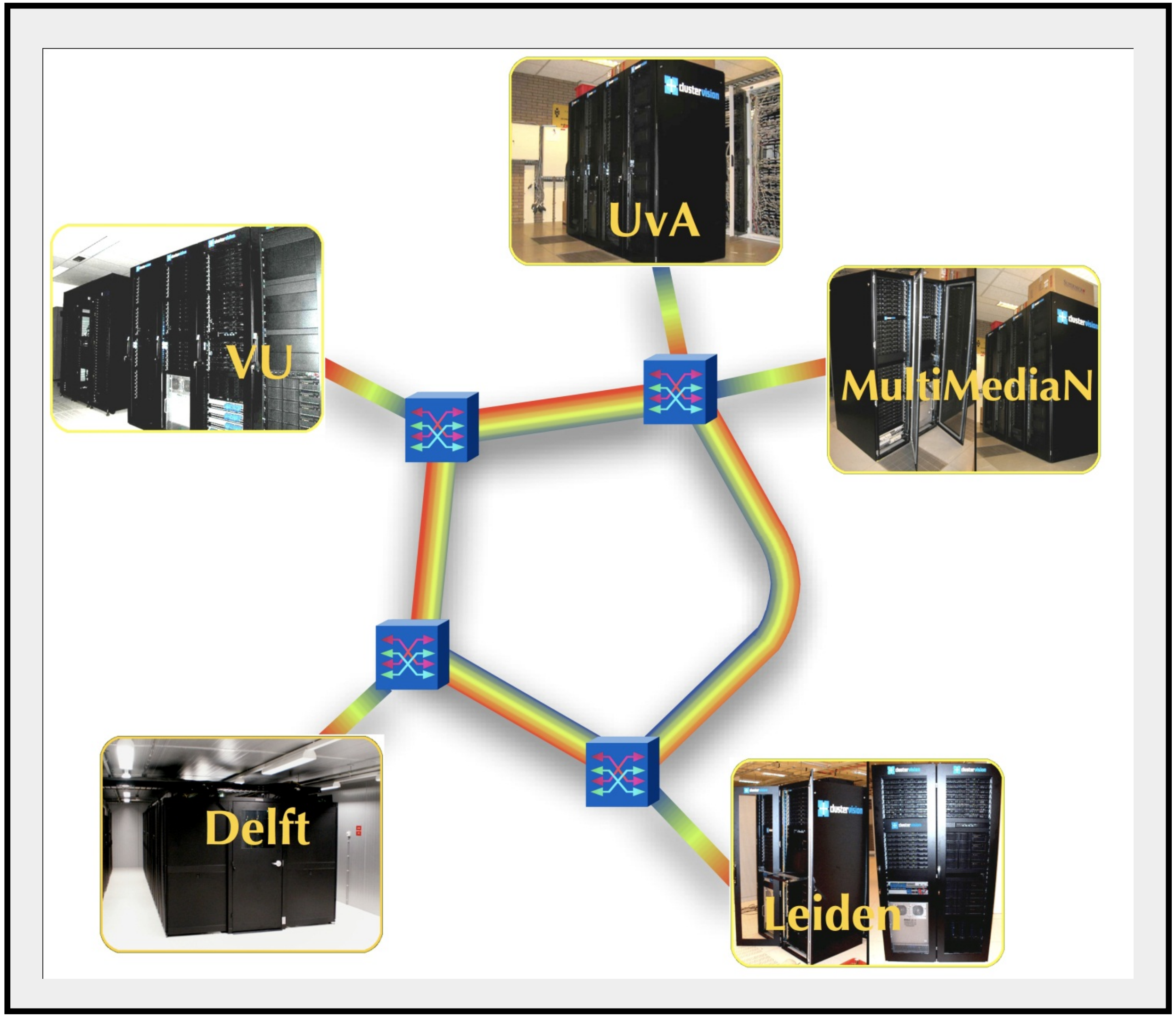
- Ibis – [IEEE Computer 2010]
- StarPlane
- Wide-area Awari
- Distributed Model Checking



vrije Universiteit

Netherlands Organisation for Scientific Research

StarPlane



vl·e

GigaPort

MultimediaN

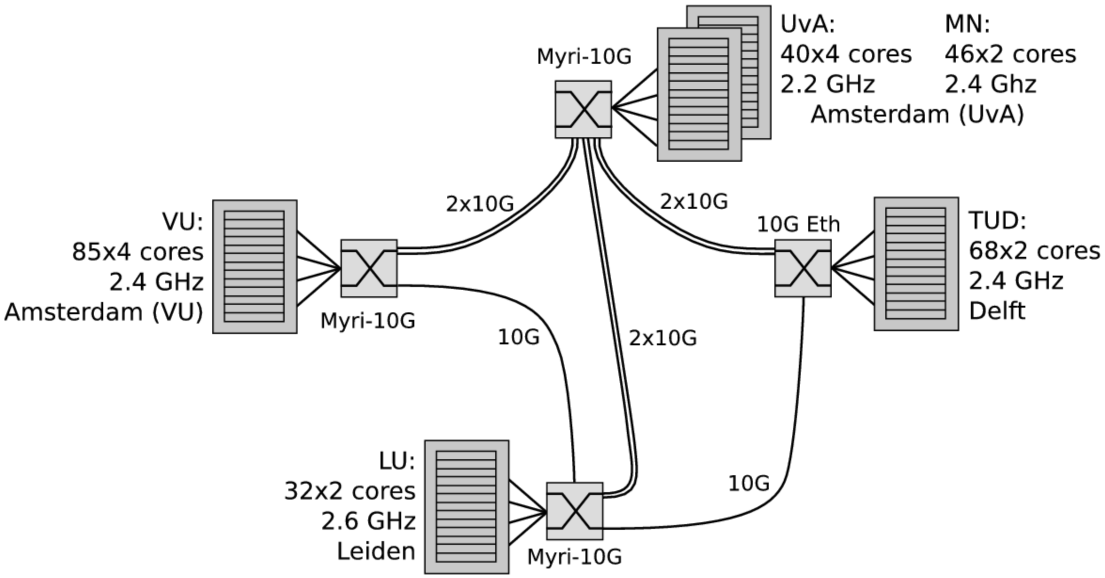
Advanced School for Computing and Imaging

vrije Universiteit

GigaPort

StarPlane

- Multiple dedicated 10G light paths between sites
- Idea: dynamically change wide-area topology



vrije Universiteit

Wide-area Awari

- Based on retrograde analysis
  - Backwards analysis of search space (database)
- Partitions database, like transposition tables
  - Random distribution good load balance
- Repeatedly send results to parent nodes
  - Asynchronous, combined into bulk transfers
- Extremely communication intensive:
  - 1 Pbit of data in 51 hours (on 1 DAS-2 cluster)

vrije Universiteit

Awari on DAS-3 grid

- Implementation on single big cluster
  - 144 cores
  - Myrinet (MPI)
- Naïve implementation on 3 small clusters
  - 144 cores
  - Myrinet + 10G light paths (OpenMPI)

vrije Universiteit

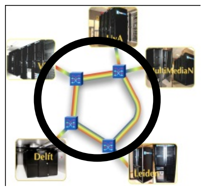
Initial insights

- Single-cluster version has high performance, despite high communication rate
  - Up to 28 Gb/s cumulative network throughput
- Naïve grid version has flow control problems
  - Faster CPUs overwhelm slower CPUs with work
  - Unrestricted job queue growth
    - ➔ Add regular global synchronizations (barriers)

vrije Universiteit

Optimizations

- Scalable barrier synchronization algorithm
  - Ring algorithm has too much latency on a grid
  - Tree algorithm for barrier&termination detection



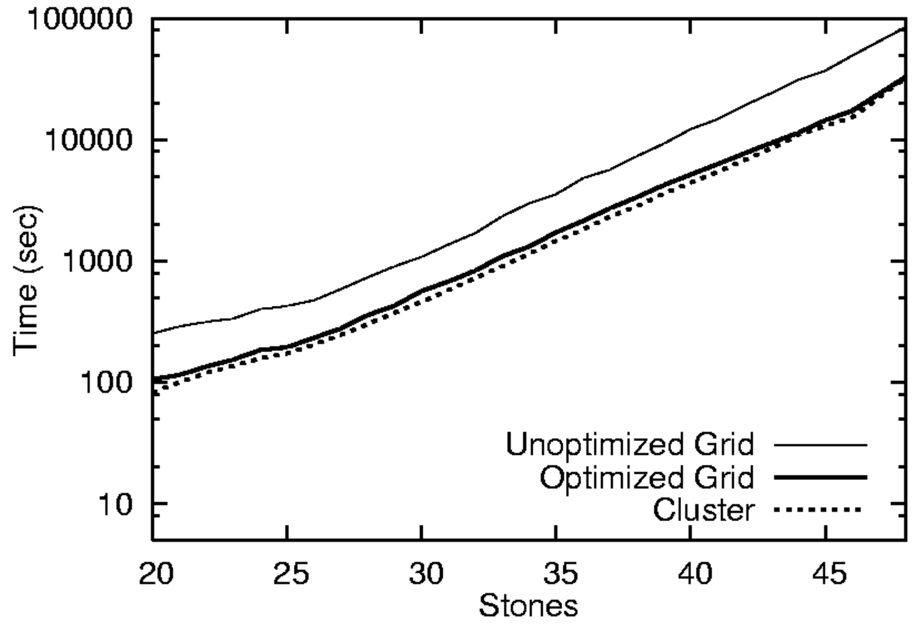
- Reduce host overhead
  - CPU overhead for MPI message handling/polling
- Optimize grain size per network (LAN vs. WAN)
  - Large messages (much combining) have lower host overhead but higher load-imbalance

➢ [CCGrid 2008]

vrije Universiteit

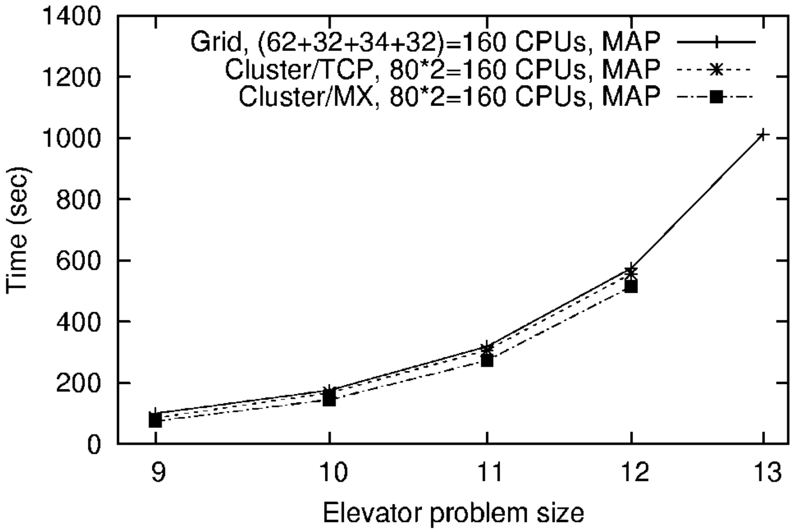
Performance

- Optimizations improved grid performance by 50%
- Grid version only 15% slower than 1 big cluster
  - Despite huge amount of communication (14.8 billion messages for 48-stone database)

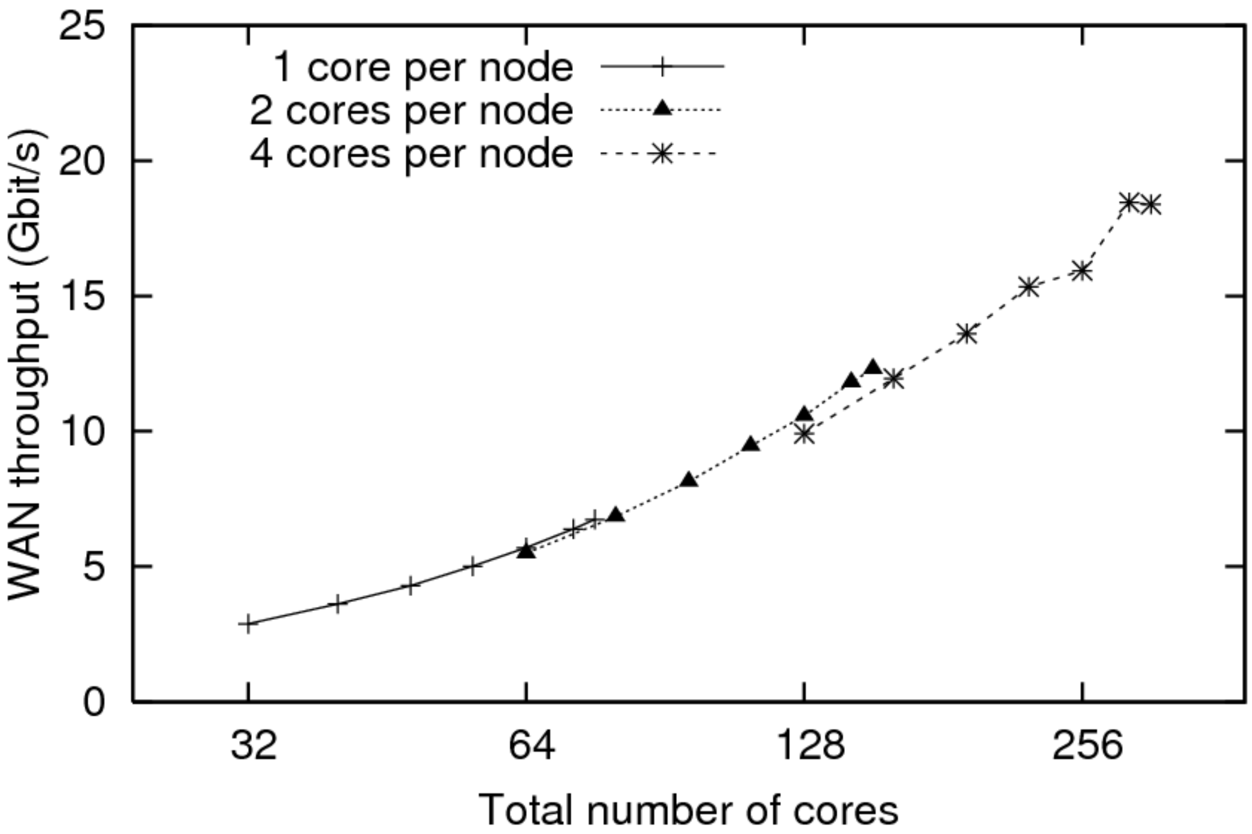


From Games to Model Checking

- Distributed model checking has very similar communication pattern as Awari
  - Search huge state spaces, random work distribution, bulk asynchronous transfers
- Can efficiently run DiVinE model checker on wide-area DAS-3, use up to 1 TB memory [IPDPS'09]



Required wide-area bandwidth



DAS-4 research examples

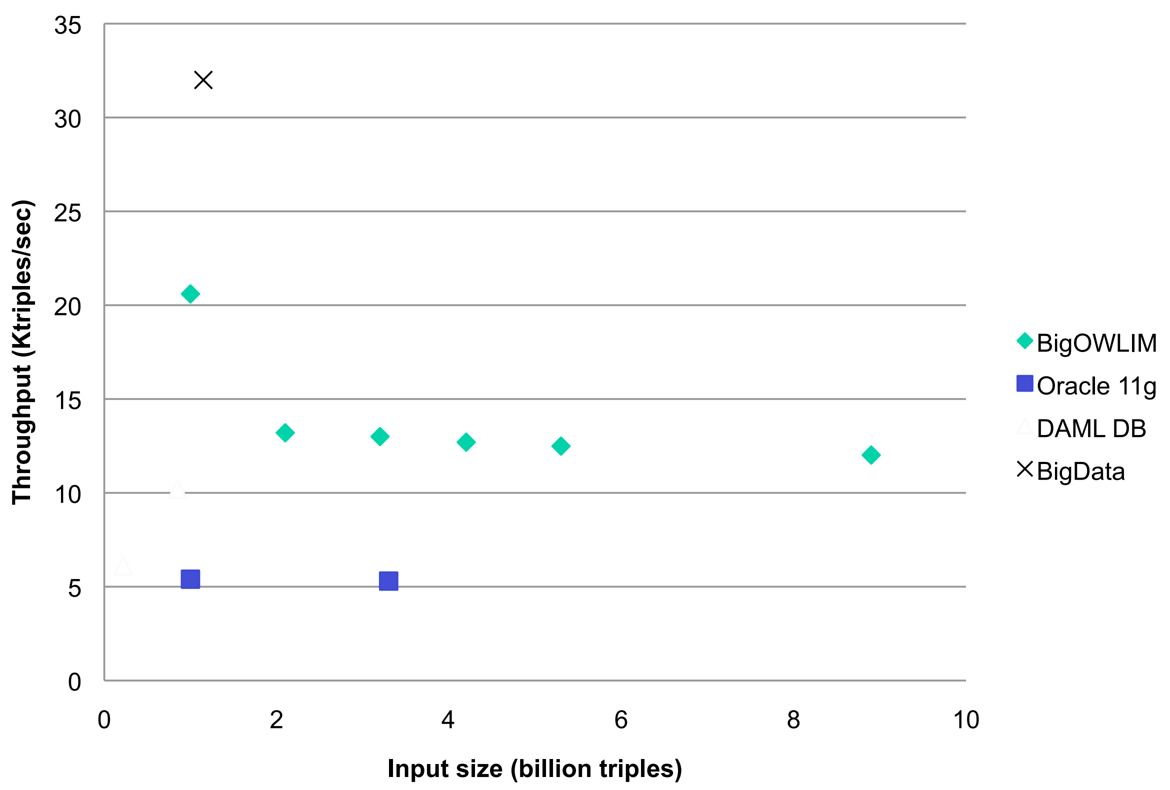
- Distributed reasoning
  - (Jacopo Urbani)
- Multimedia analysis on GPUs
  - (Ben van Werkhoven)

vrije Universiteit

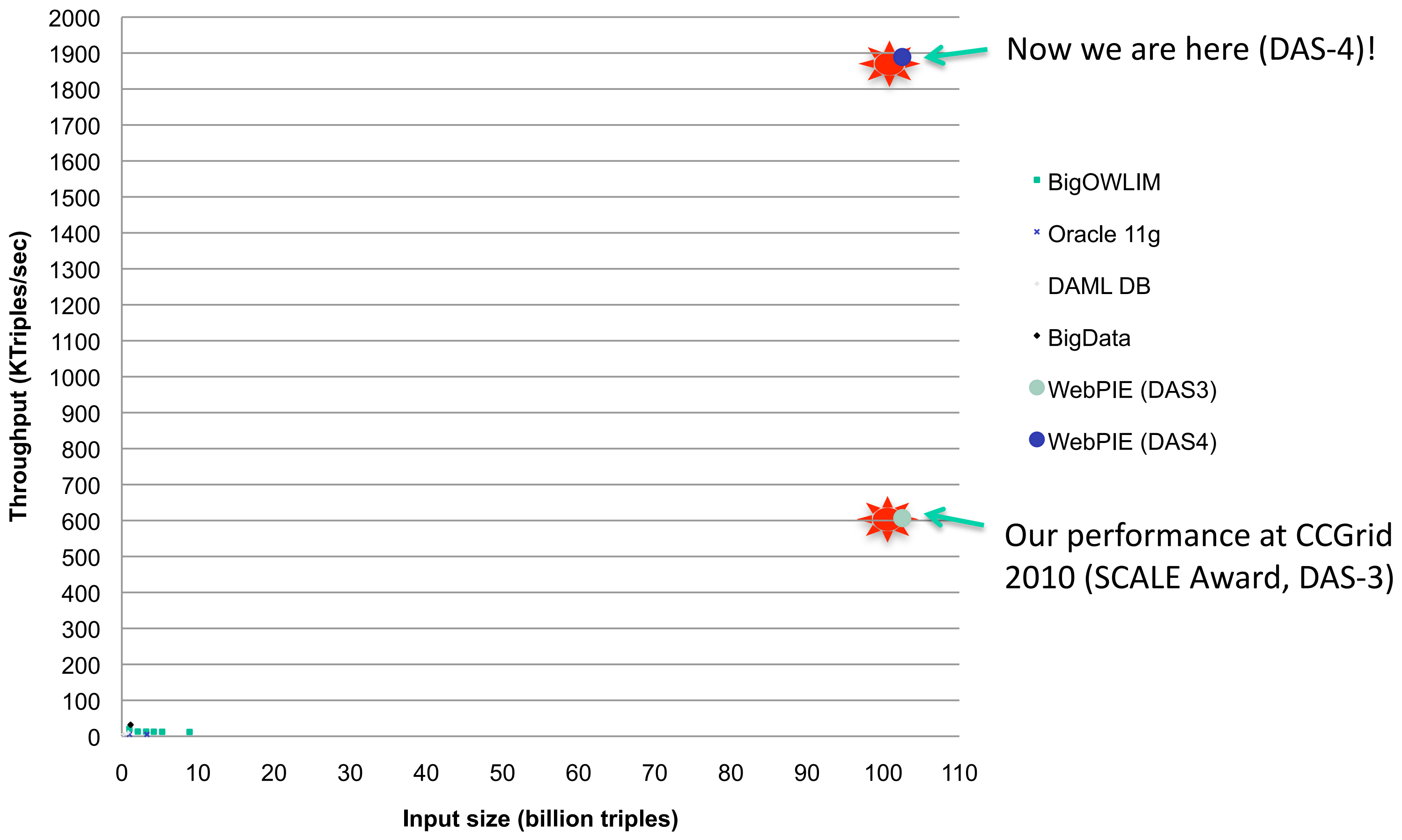
WebPIE

- The Semantic Web is a set of technologies to enrich the current Web
- Machines can reason over SW data to find best results to the queries
- WebPIE is a MapReduce reasoner with linear scalability
- It significantly outperforms other approaches by one/two orders of magnitude

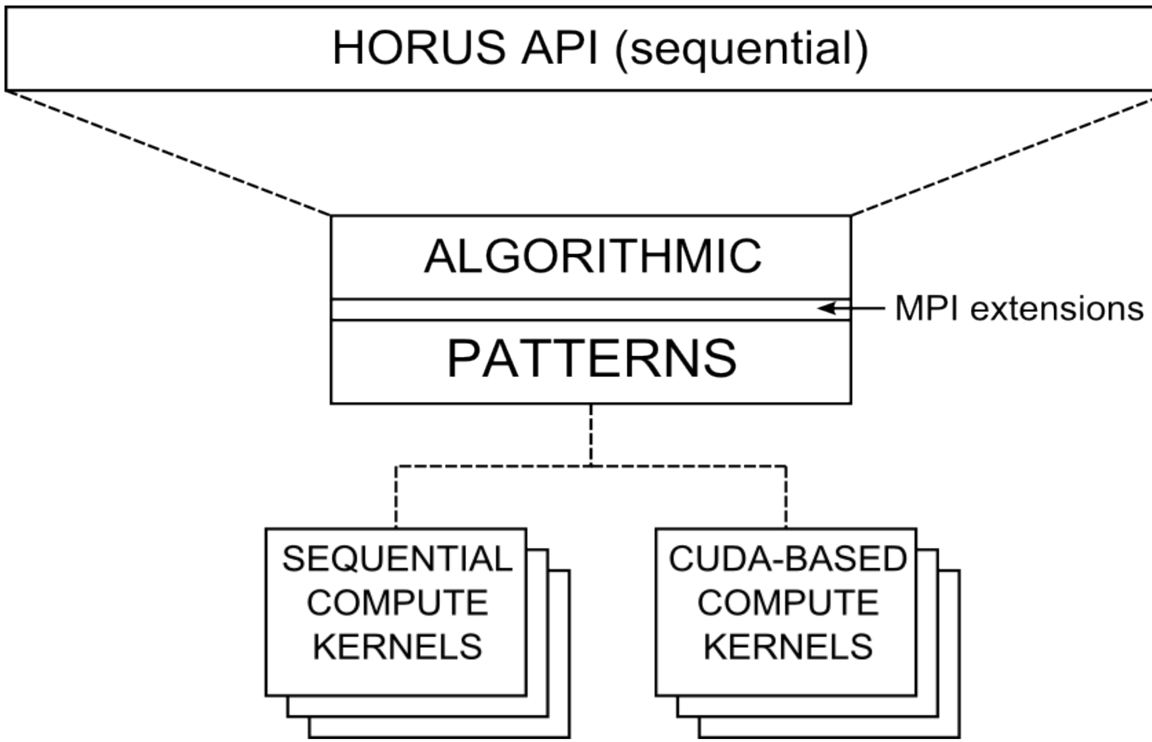
Performance previous state-of-the-art



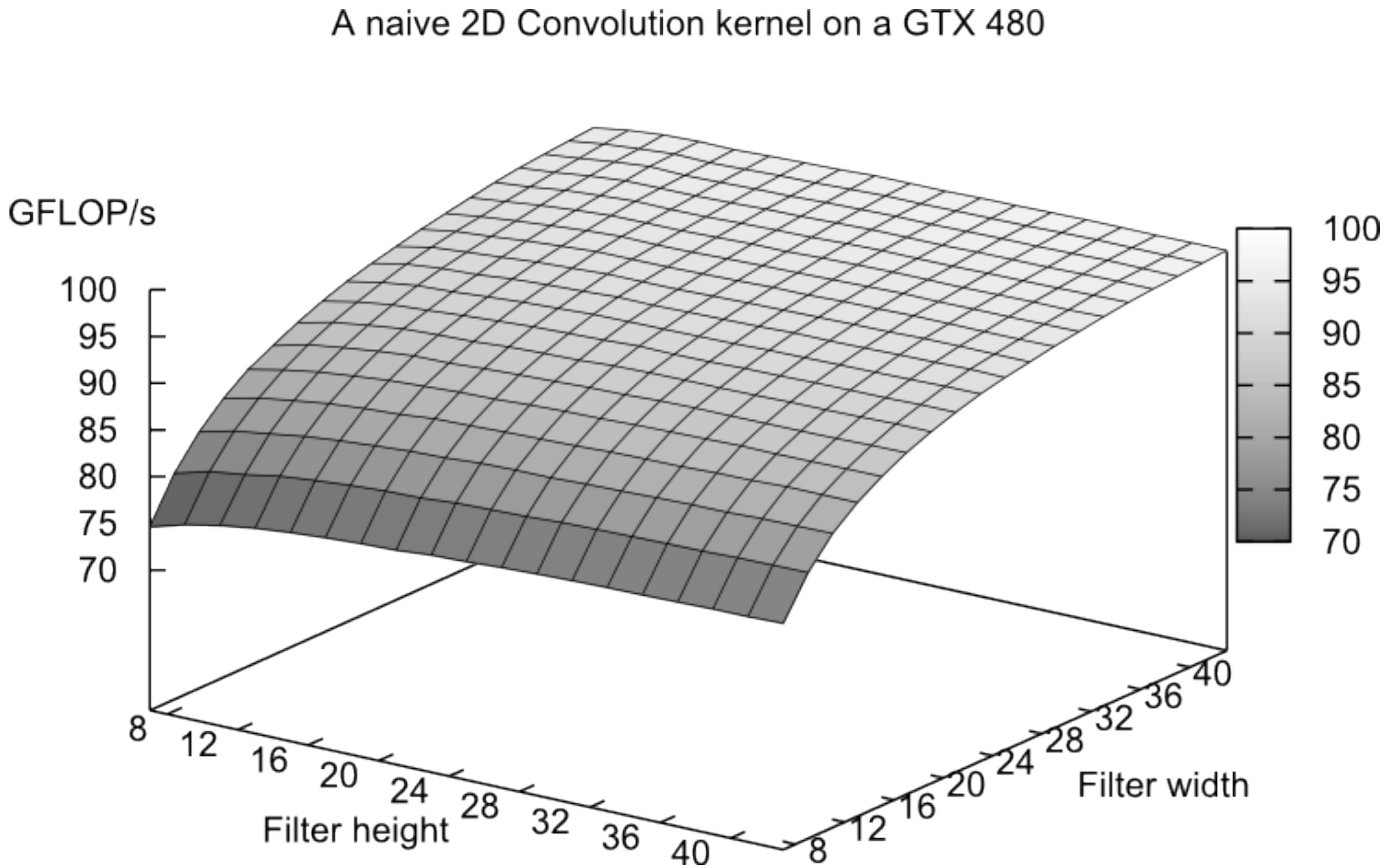
Performance WebPIE



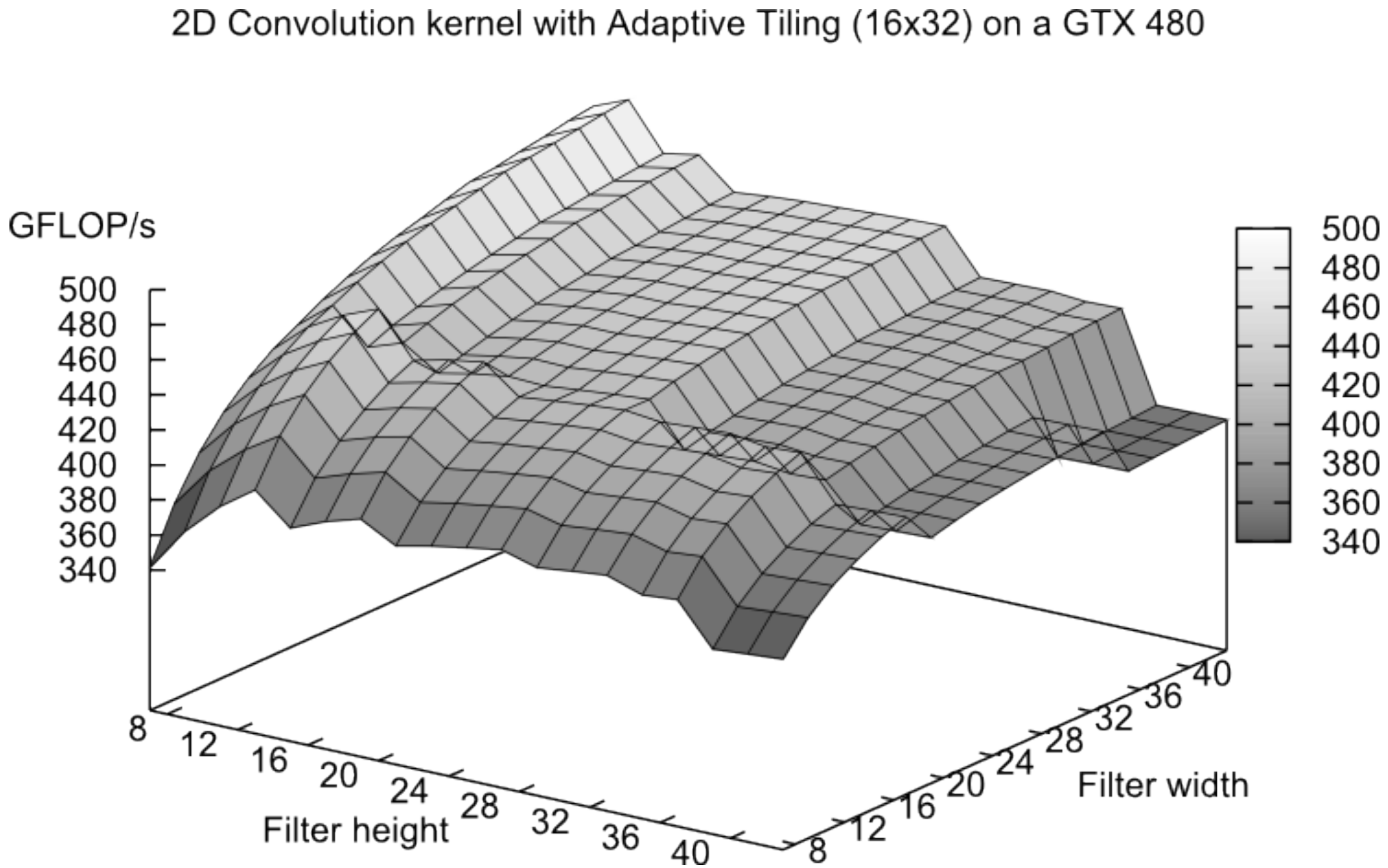
Parallel-Horus: User Transparent Parallel Multimedia Computing on GPU Clusters



Naive 2D Convolution Kernel on DAS-4 GPU



Our best performing 2D Convolution Kernel



Conclusions

- Having a dedicated distributed infrastructure for CS enables experiments that are impossible on production systems
  - Need long-term organization (like ASCI)
- DAS has had a huge impact on Dutch CS, at moderate cost
  - Collaboration around common infrastructure led to large joint projects (VL-e, SURFnet) & grants

Acknowledgements

VU Group:
Niels Drost
Ceriel Jacobs
Roelof Kemp
Timo van Kessel
Thilo Kielmann
Jason Maassen
Rob van Nieuwpoort
Nick Palmer
Kees van Reeuwijk
Frank Seinstra
Jacopo Urbani
Kees Verstoep
Ben van Werkhoven
& many others

DAS Steering Group:
Lex Wolters
Dick Epema
Cees de Laat
Frank Seinstra
John Romein
Rob van Nieuwpoort

DAS management:
Kees Verstoep

DAS grandfathers:
Andy Tanenbaum
Bob Hertzberger
Henk Sips

Funding:
NWO, NCF, SURFnet, VL-e, MultimediaN

vrije Universiteit