Twenty years of Grid Scheduling and Beyond

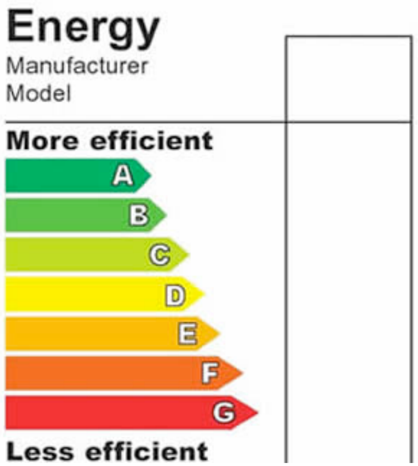
12th IEEE/ACM Symposium on Custer, Cloud and Grid Computing
Ottawa, Canada

Dick Epema

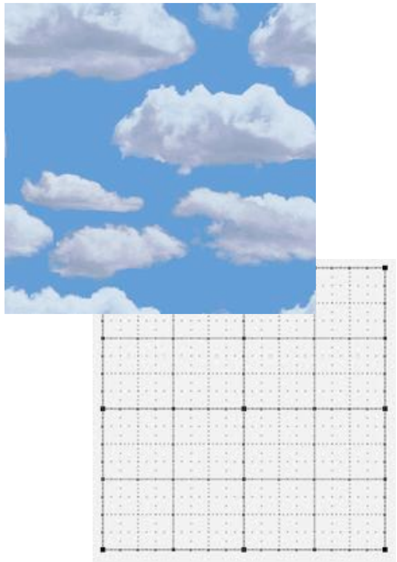
Parallel and Distributed Systems Group
Delft University of Technology
Delft, the Netherlands
and
System Architecture and Networking Group
Eindhoven University of Technology
Eindhoven, the Netherlands

May 16, 2012

Overview



42u Server Rack



Tulips in Ottawa

Every year 10,000 tulip bulbs

Dutch royal family (later queen) in Ottawa in WWII

Liberation of the Netherlands by the Canadians, May 1945

Condor (1/7): my first grid computing

- **Condor**
  - is a **high-throughput** scheduling system
  - started around 1986 as one of many **batch queuing systems** for clusters (of desktop machines), and **has survived!**
  - supports **cycle scavenging**: use idle time on clusters of machines
  - introduced the notions of **matchmaking** and **classads**
  - provides remote system calls, a queuing mechanism, scheduling policies, priority scheme, resource monitoring
  - initiated and still being developed by Miron Livny, Madison, Wisc.

D.H.J. Epema, M. Livny, R. van Dantzig, X. Evers, and J. Pruyne, "A Worldwide Flock of Condors: Load Sharing among Workstation Clusters," *Future Generation Computer Systems*, Vol. 12, pp. 53-65, 1996.

Condor (2/7): matchmaking

- **Basic operation of Condor:**

  1a **jobs** send **classads** to the **matchmaker**

  1b **machines** send **classads** to the **matchmaker**

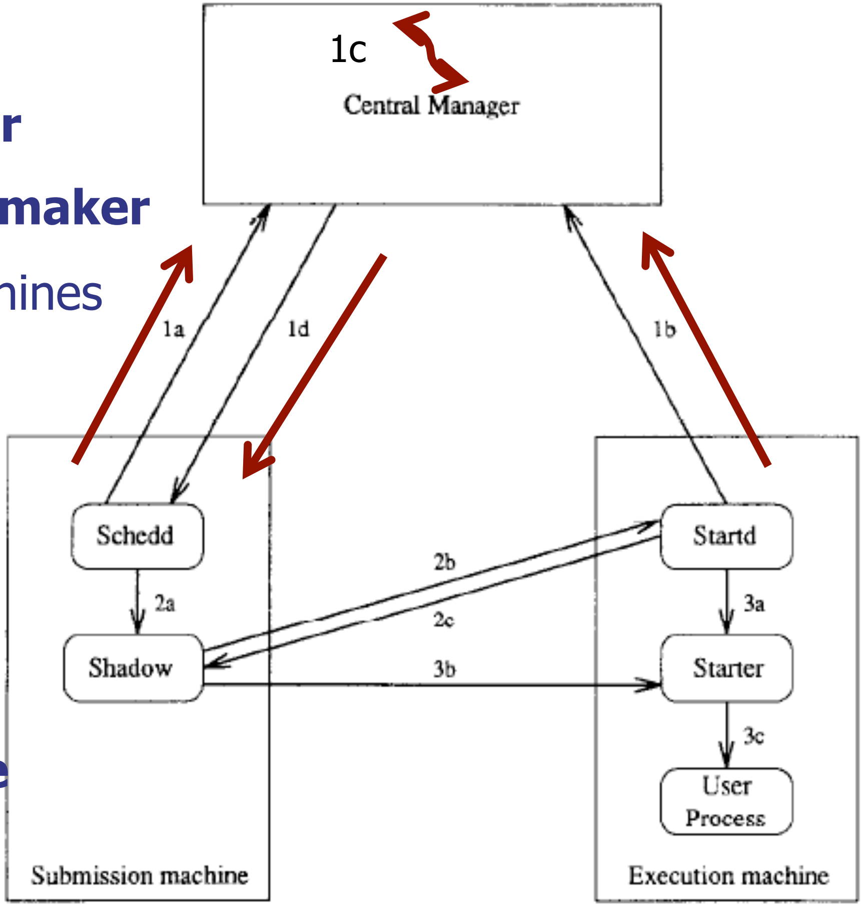
  1c the matchmaker **matches** jobs and machines

  1d and notifies the **submission machine**

  2a which starts a **shadow** process is that represents the remote job on the execution machine

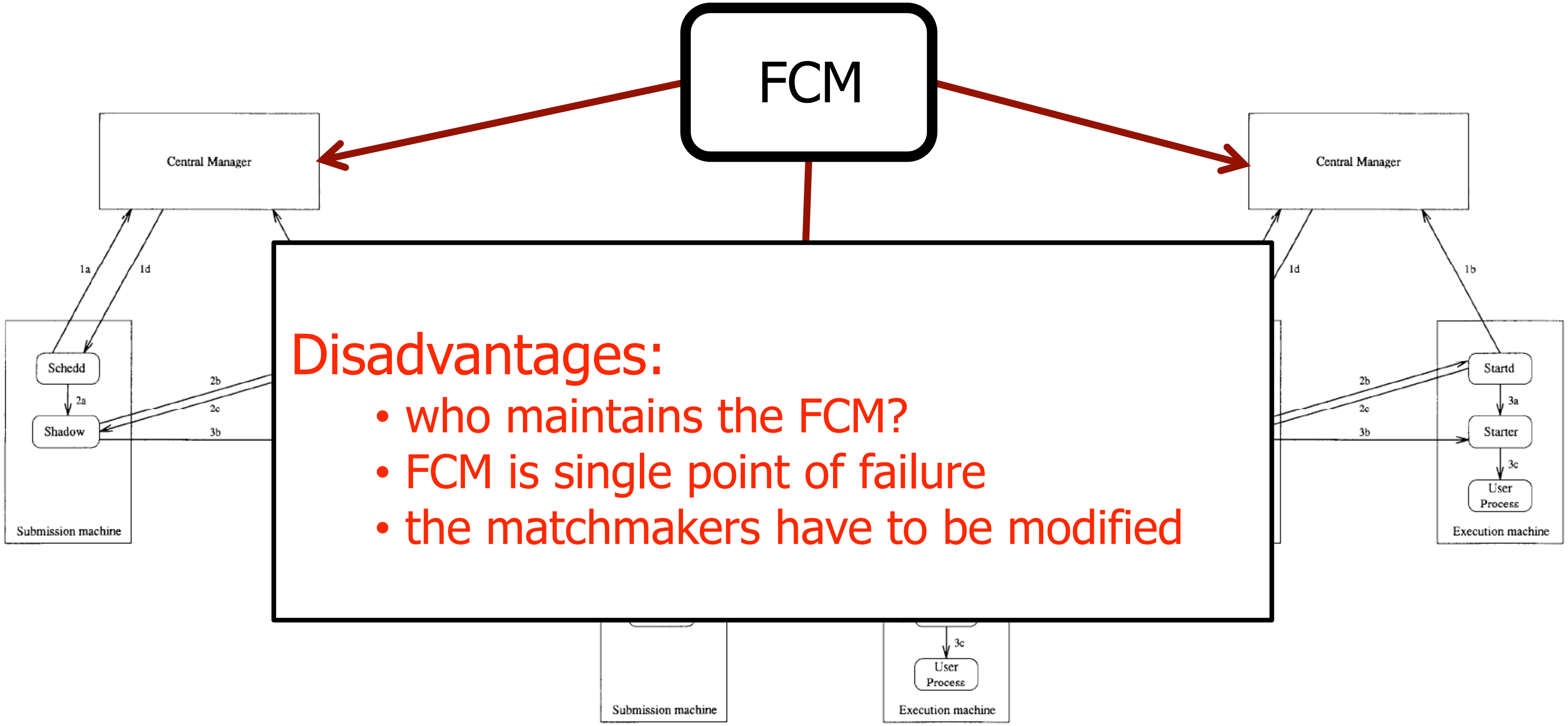
  2b/c and contacts the **execution machine**

  3b/c on the execution machine, the actual **remote user job** is started



Condor (3/7): combining pools (design 1)

- Federation with a **Flock Central Manager**:

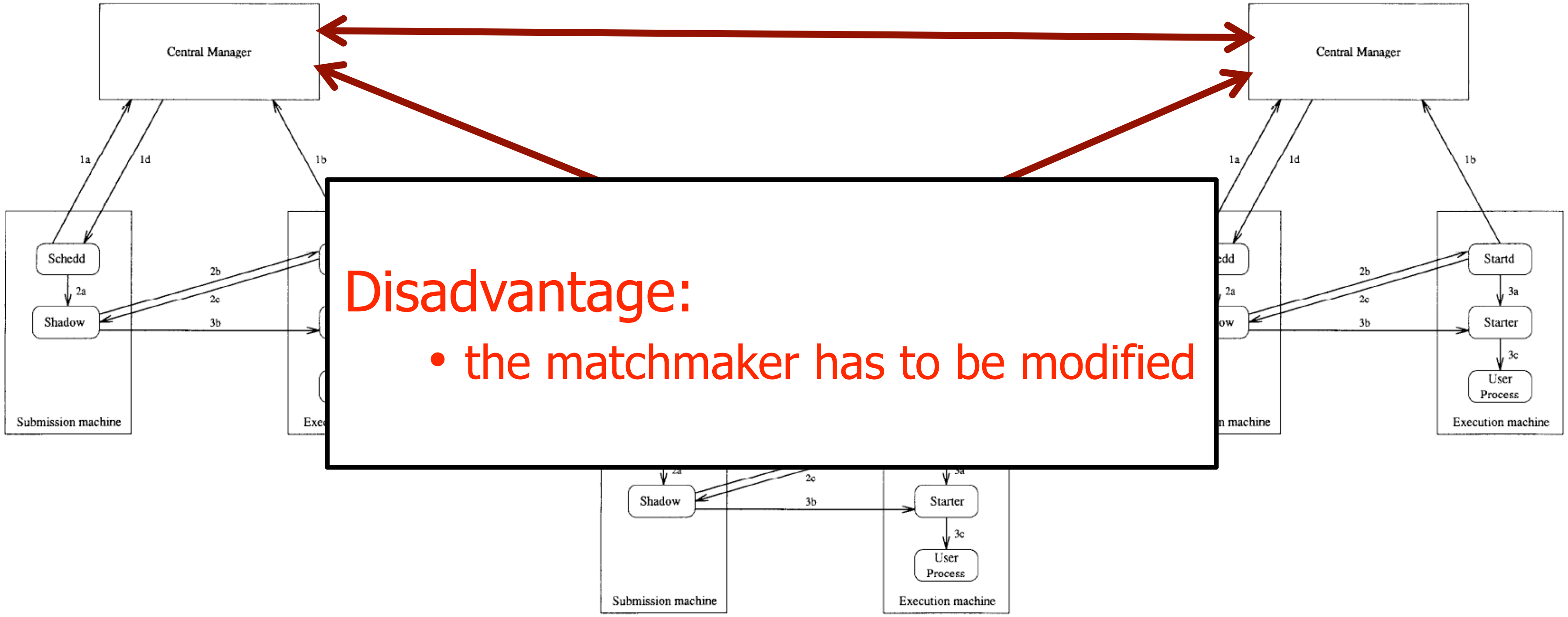


Disadvantages:

- who maintains the FCM?
- FCM is single point of failure
- the matchmakers have to be modified

Condor (4/7): combining pools (design 2)

- Protocol **between Central Managers**:

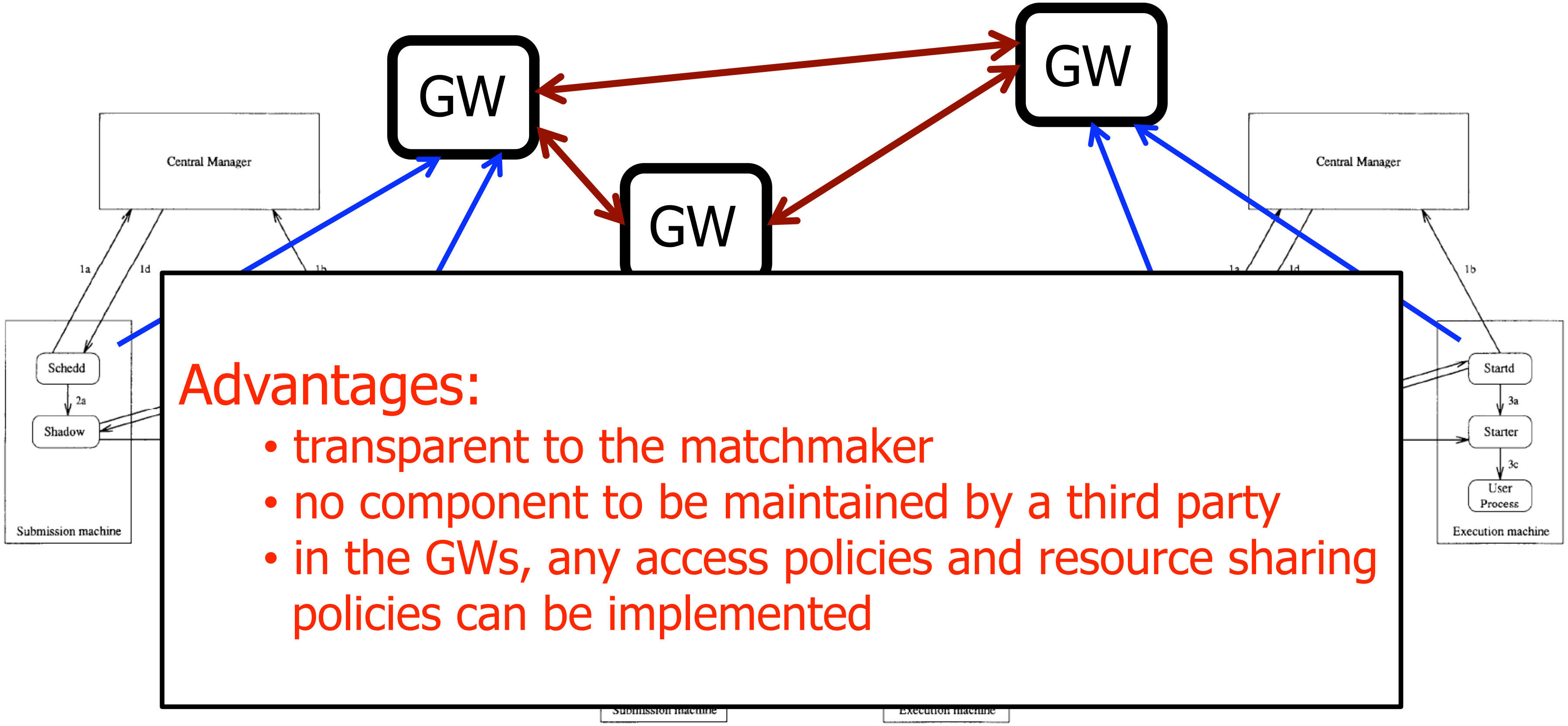


Disadvantage:

- the matchmaker has to be modified

Condor (5/7): combining pools (design 3)

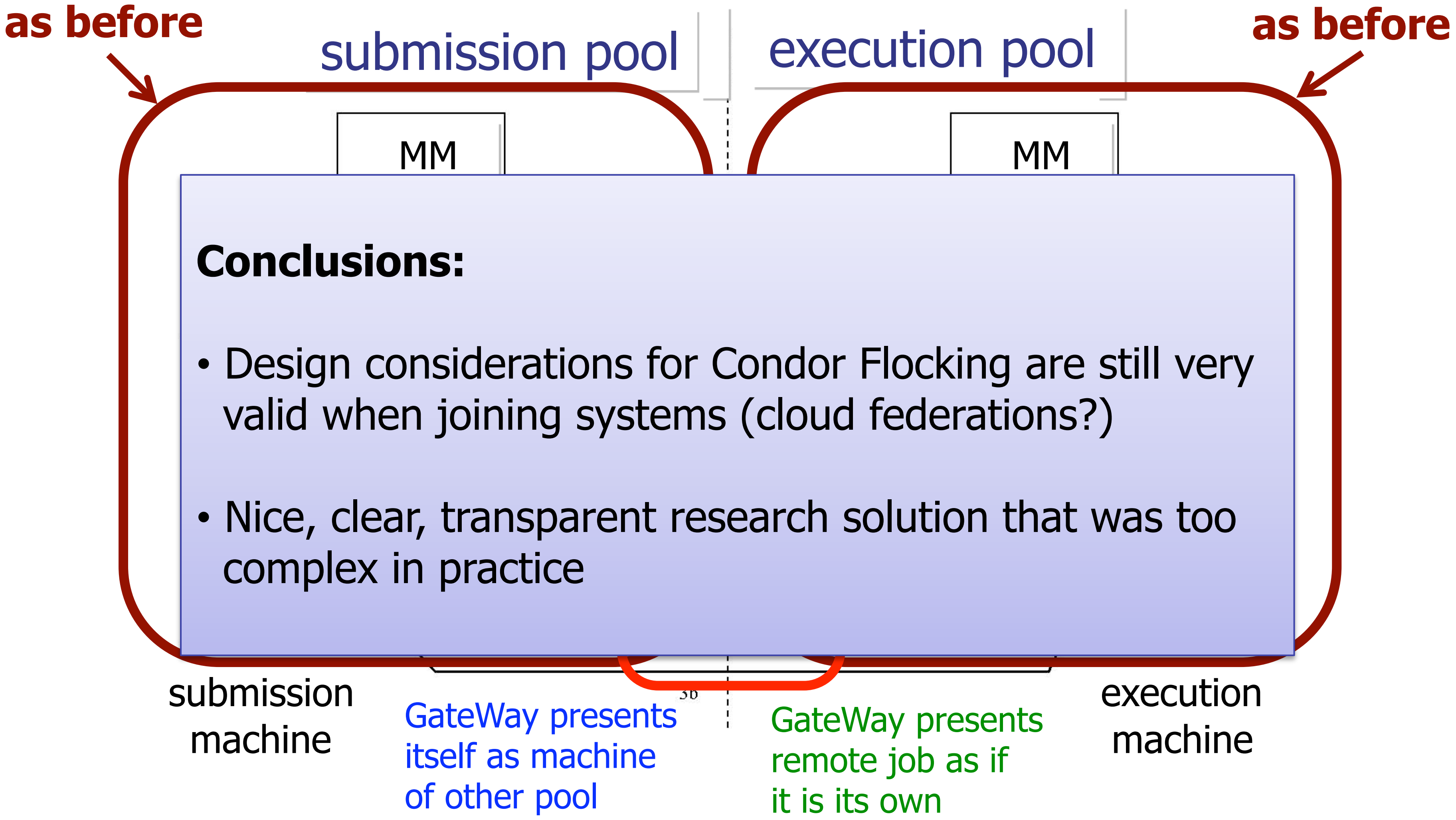
- Connecting pools **network-style with GateWays**: **Condor flocking**



Advantages:

- transparent to the matchmaker
- no component to be maintained by a third party
- in the GWs, any access policies and resource sharing policies can be implemented

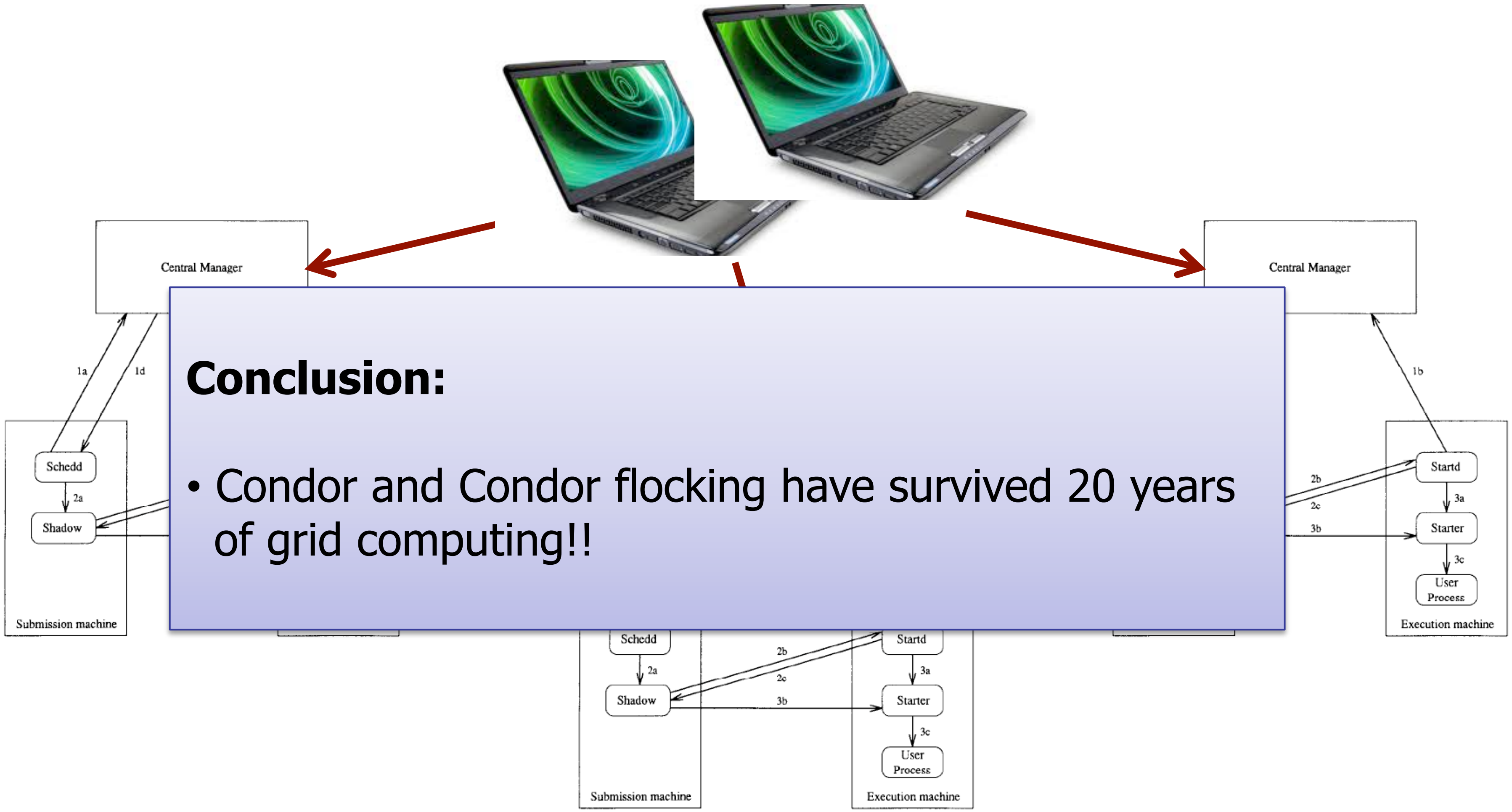
Condor (6/7): flocking


as before
submission pool
execution pool
as before
MM
MM
submission machine
GateWay presents itself as machine of other pool
GateWay presents remote job as if it is its own
execution machine

Conclusions:

- Design considerations for Condor Flocking are still very valid when joining systems (cloud federations?)
- Nice, clear, transparent research solution that was too complex in practice

Condor (7/7): user flocking

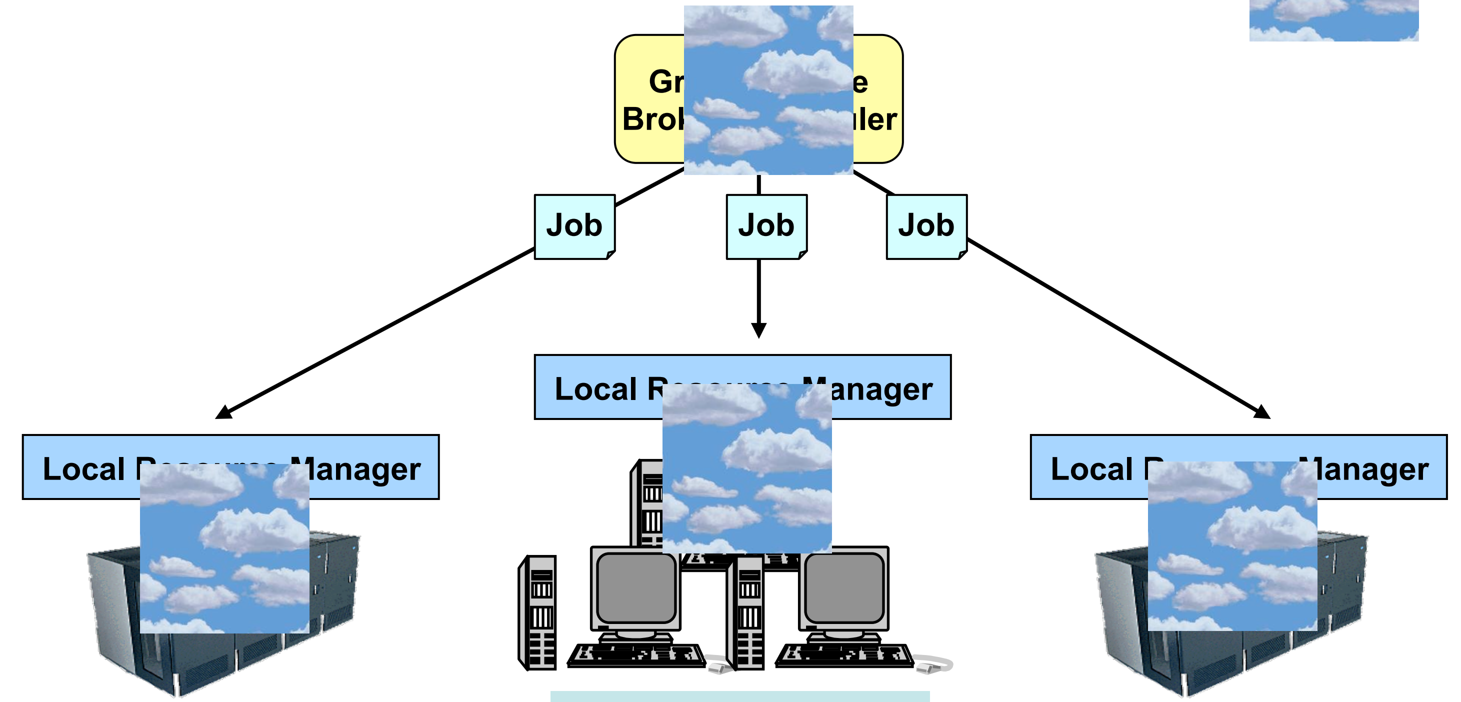


Conclusion:

- Condor and Condor flocking have survived 20 years of grid computing!!

H[illegible] to select resources in the [illegible]

- [illegible] **scheduling** is the process of assigning jobs to [illegible] resources



Gr[illegible]e
Brok[illegible]ler

Job

Job

Job

Local R[illegible]anager

Local [illegible]anager

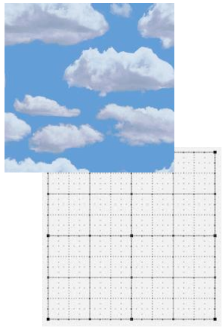
Local [illegible]anager

Clusters
(Space Shared Allocation)

Single CPU
(Time Shared Allocation)

Clusters
(Space Shared Allocation)

Resource Characteristics in Grids (1)



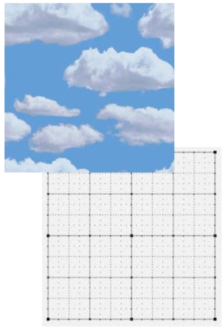
- **Autonomous**
  - each resource has its own management policy or scheduling mechanism
  - no central control/**multi-organizational** sets of resources

+

- **Heterogeneous**
  - hardware (processor architectures, disks, network)
  - basic software (OS, libraries)
  - grid software (middleware)
  - systems management (security set-up, runtime limits)

-/+

Resource Characteristics in Grids (2)



- **Size**
  - large numbers of nodes, providers, consumers
  - large amounts of data

+

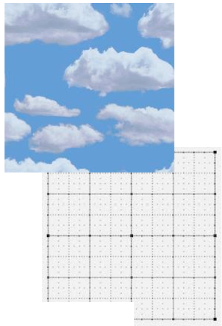
- **Varying Availability**
  - resources can join or leave to the grid at any time due to maintenance, policy reasons, and failures

-/+

- **Insecure and unreliable environment**
  - prone to various types of attacks

?

Problems in Grid Scheduling (1)



1. **Grid schedulers do not own resources themselves**
   - they have to negotiate with autonomous local schedulers
   - authentication/multi-organizational issues

=

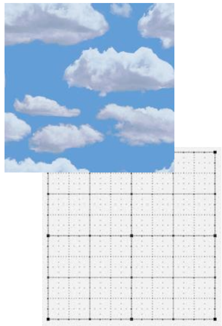
2. **Grid schedulers have to interface to different local schedulers**
   - some may have support for reservations, others are queuing
   - some may support checkpointing, migration, etc

=

3. **Structure of applications**
   - many different structures (parallel, PSAs, workflows, database, etc.)
   - need for application adaptatio es

ever more support

Problems in Grid scheduling (2)



4. **Lack of a reservation mechanism**
   - but with such a mechanism we need good runtime e

-

5. **Heterogeneity**
   - see above

6. **Failures**
   - monitor the progress of applications/sanity of systems
   - only thing we know to do upon failures: (move and) rest (possibly from a checkpoint)

?

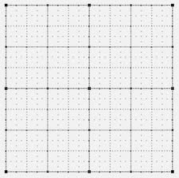
7. **Performance metric**
   - turn-around time

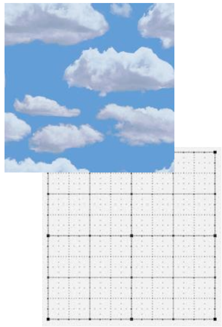
cost

8. **Reproducibility of performance experiments**

+/-

Grids versus Clouds





heterogeneous	homogeneous
many types of systems	datacenters
multi-organizational	[illegible]
free (?)	payment model
	energy awareness
----------------------------- +	**-------------------------------- +**
Grids	**Clouds**

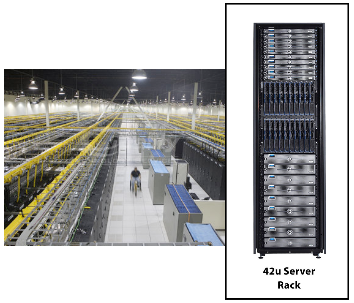
Conclusion?:

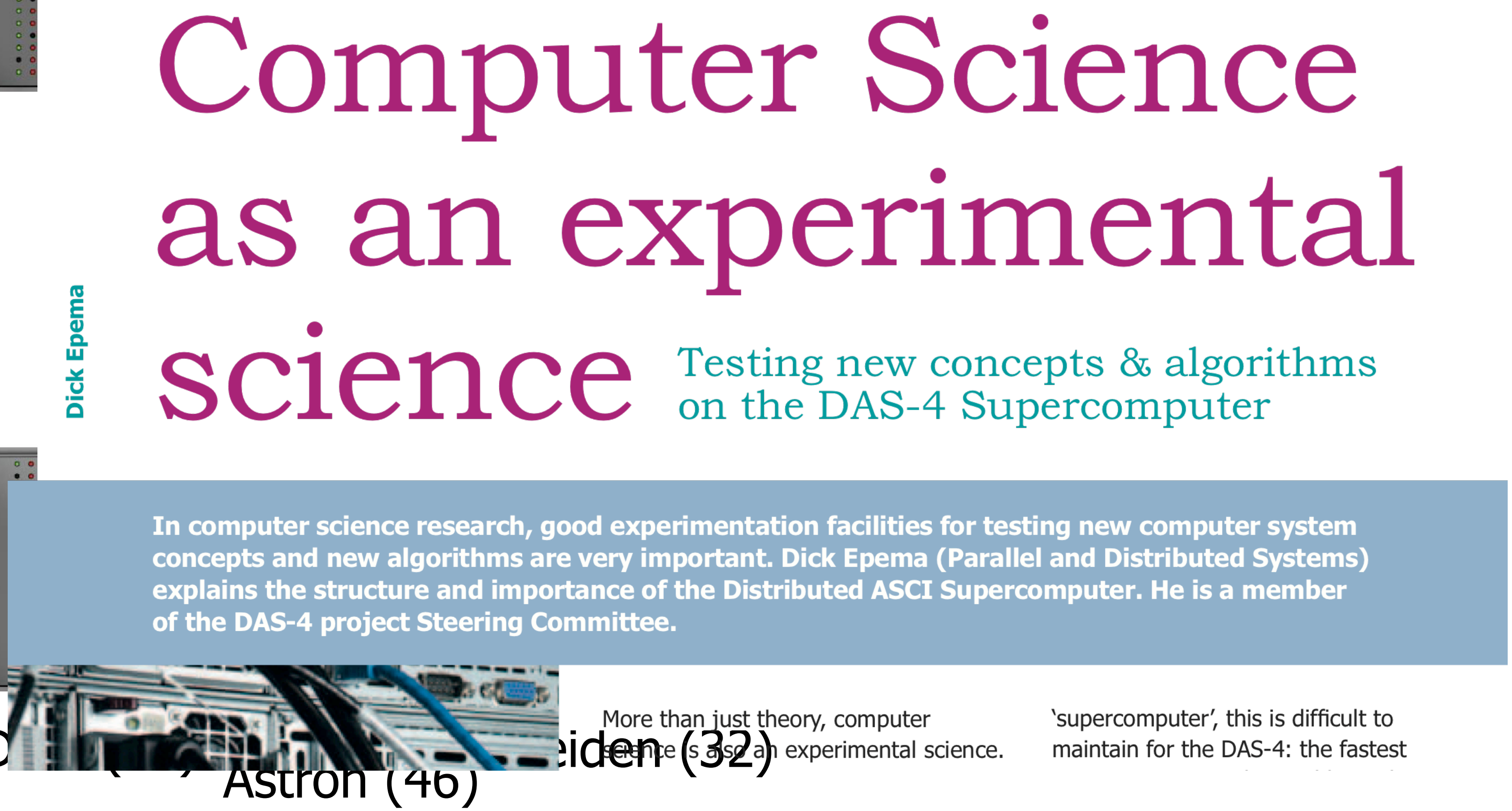
- Aren't clouds are just the next variety of distributed systems (just like grids previously)?

Experimentation (1): DAS-4

VU (148 CPUs)

UvA/MultimediaN (72)





Computer Science as an experimental science

Testing new concepts & algorithms on the DAS-4 Supercomputer

Dick Epema

In computer science research, good experimentation facilities for testing new computer system concepts and new algorithms are very important. Dick Epema (Parallel and Distributed Systems) explains the structure and importance of the Distributed ASCI Supercomputer. He is a member of the DAS-4 project Steering Committee.

More than just theory, computer science is also an experimental science.

'supercomputer', this is difficult to maintain for the DAS-4: the fastest

TU D

Astron (46)

iden (32)

arch

0

May 16, 2012

Advanced School for Computing and Imaging

NWO Netherlands Organisation for Scientific Research

TU Delft

Experimentation (2): scale

- When the DAS2 started, it entered the TOP 500
- Top500 list of November 2011:

	Number of cores
#1	705,024
#42 Amazon Web Services	17,024
#483	2,048
#500	7,236
DAS-4	1,600

- What is the value of our experiments (**scale does matter**)?

KOALA: a co-allocating grid scheduler

- **Original goals:**

  1. processor co-allocation: parallel applications

  2. data co-allocation: job affinity based on data locations

  3. load sharing: in the absence of co-allocation

  while being transparant for local schedulers

- **Additional goals:**
  - **research vehicle** for grid and cloud research
  - support for (other) popular application types

- **KOALA**
  - is written in Java
  - is middleware independent (initially Globus-based)
  - **has been deployed** on the DAS2 - DAS4 since sept 2005

KOALA: the runners

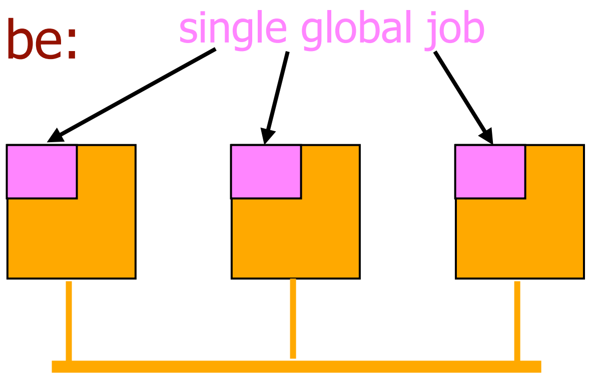
- The KOALA **runners** are **adaptation modules** for different application types:
  - set up communication / name server / environment
  - la
  - sc
- **Curr**
  - **C**
  - **I** y
  - **M**
  - **O** ations
  - **W**
  - **MR-runner:** for **MapReduce** applications (under construction)

Conclusion:

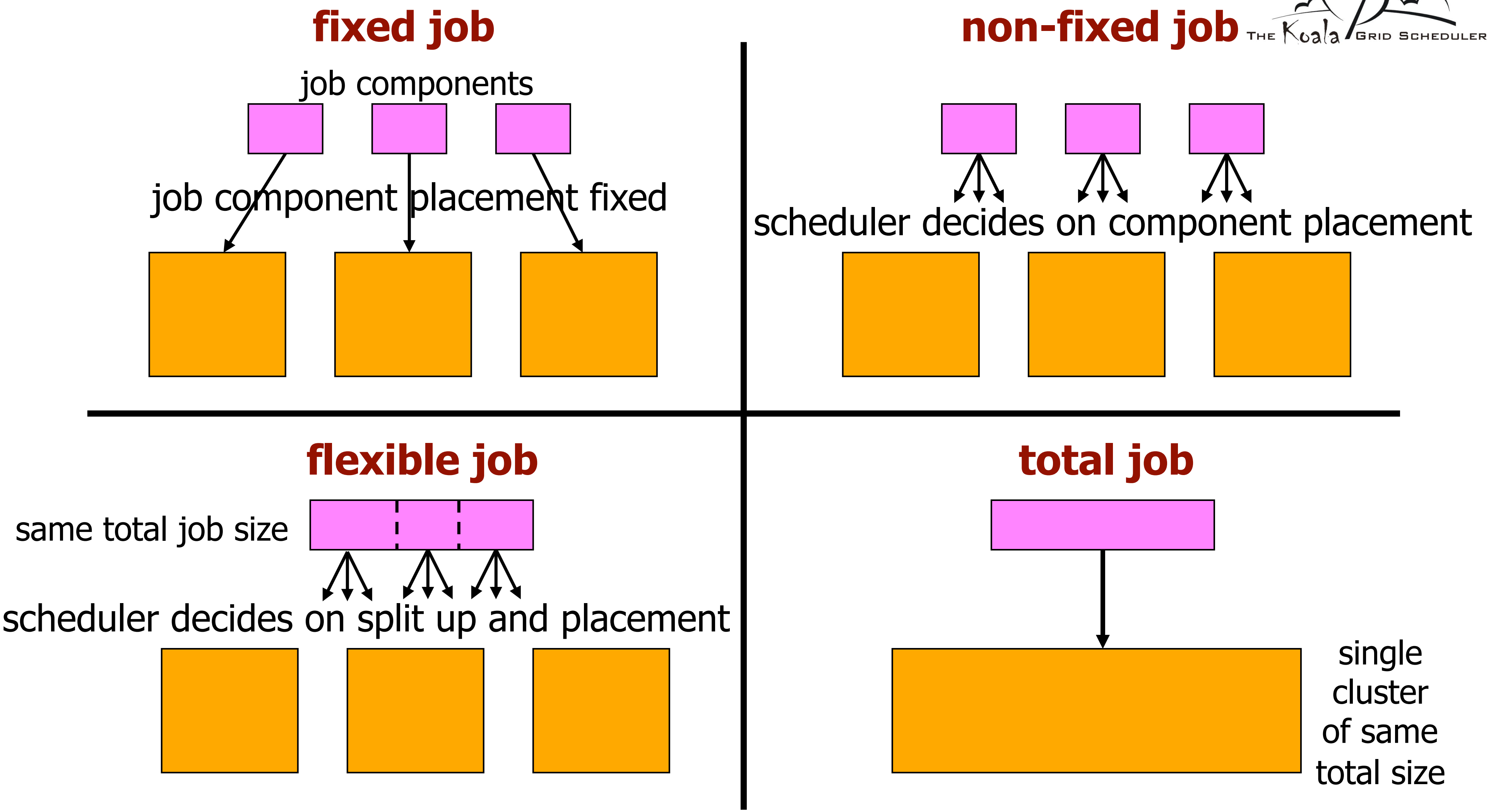
- Very beneficial to have a deployed research vehicle (DAS4 + KOALA) for
  - driving research
  - doing experimentation
  - visibility

Co-Allocation (1)

- In grids, jobs may use resources in multiple sites: **co-allocation** or **multi-site operation**
- **Reasons**:
  - to benefit from available resources (e.g., processors)
  - to access and/or process **geographically spread data**
  - application characteristics (e.g., simulation in one location, visualization in another)
- **Resource possession in different sites** can be:
  - simultaneous (e.g., parallel applications)
  - coordinated (e.g., workflows)
- **With co-allocation**:
  - more difficult **resource-discovery** process
  - need to **coordinate allocations** by autonomous resource managers



Co-allocation (2): job types



fixed job

job components

job component placement fixed

non-fixed job

scheduler decides on component placement

flexible job

same total job size

scheduler decides on split up and placement

total job

single cluster of same total size

Co-allocation (3): slowdown

- Co-allocated applications are **less efficient** due to the relatively **slow wide-area communications**
- **Slowdown of a job**:

$$\frac{\text{execution time on multicluster}}{\text{execution time on single cluster}}$$

(>1 usually)

- Processor co-allocation is a **trade-off** between
  - **faster access to more capacity**, and higher utilization
  - **shorter execution times**

Co-allocation (4): scheduling policies

- **Placement policies** dictate where the components of a job go
- **Placement policies for non-fixed jobs**:
  1. **Load-aware**:
     (balance load in clusters)
     Worst Fit (**WF**)
  2. **Input-file-location-aware**:
     (reduce file-transfer times)
     Close-to-Files (**CF**)
  3. **Communication-aware**:
     (reduce number of wide-area messages)
     Cluster Minimization (**CM**)
- **Placement policies for flexible jobs**:
  1. **Communication-aware**:
     (CM for flexible)
     Flexible Cluster Minimization (**FCM**)
  2. **Network-aware**:
     (take latency into account)
     Communication-Aware (**CA**)

Co-allocation (5): simulations/analysis

- Model has a host of parameters
-

Conclusions:

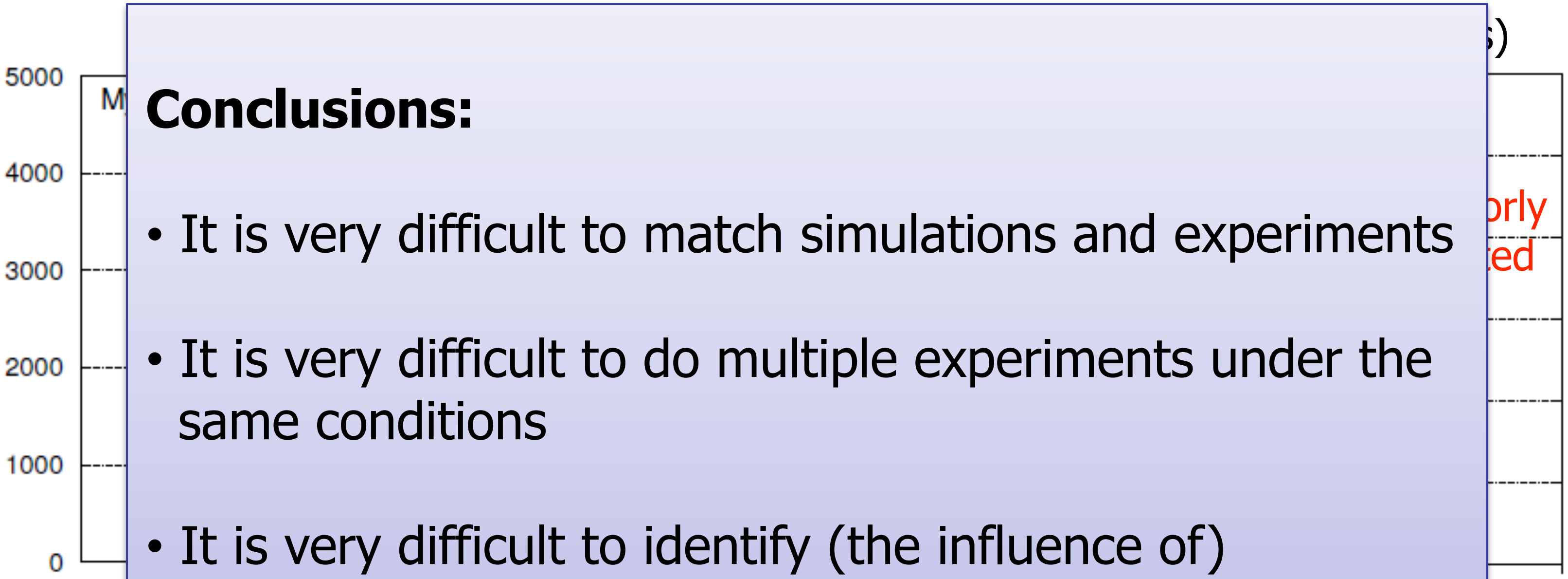
- There are fundamental problems to be derived from practical scheduling problems in grids (and clouds)
- Interplay between mathematical analysis, simulations, and experiments yields interesting results and understanding

- Mathematical analysis for maximal utilization

See, e.g.:

1. A.I.D. Bucur and D.H.J. Epema, “Trace-Based Simulations of Processor Co-Allocation Policies in Multiclusters,” *IEEE/ACM High Performance Distributed Computing (HPDC) 2003*.
2. A.I.D. Bucur and D.H.J. Epema, “Scheduling Policies for Processor Co-Allocation in Multicluster Systems," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 18, pp. 958-972, 2007.

Co-Allocation (6): Experiments on the DAS3



Conclusions:

- It is very difficult to match simulations and experiments
- It is very difficult to do multiple experiments under the same conditions
- It is very difficult to identify (the influence of) “polluting elements”

O.O. Sonmez, H.H. Mohamed, and D.H.J. Epema, “On the Benefit of Processor Co-Allocation in Multicluster Grid Systems,” *IEEE Trans. on Parallel and Distributed Systems*, Vol.21, pp. 778-789, 2010.

@large:
Massivizing Online Games as an HPC Problem

Premises:

- online gaming used to be regarded as a **multimedia topic,** but now **it is HPC**
- online gaming used to be **about networking**, but is now **all HPC**
- online gaming used to be **virtual worlds**, but is now **many applications**

What’s in a name? MSG, MMOG, …

over 250,000,000
active players in the world

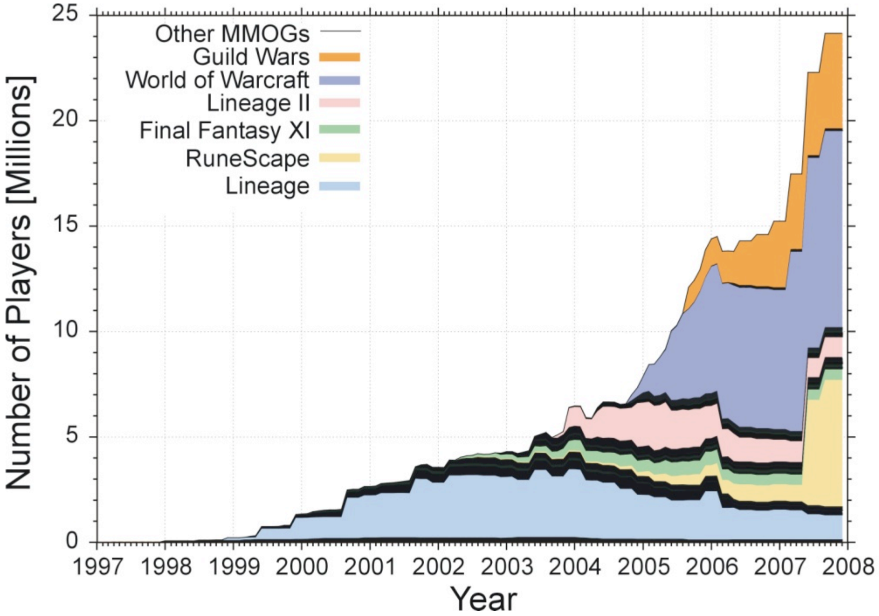
Massively Social Gaming =

(online) games with massive numbers of players (100K+), for which social interaction helps the gaming experience

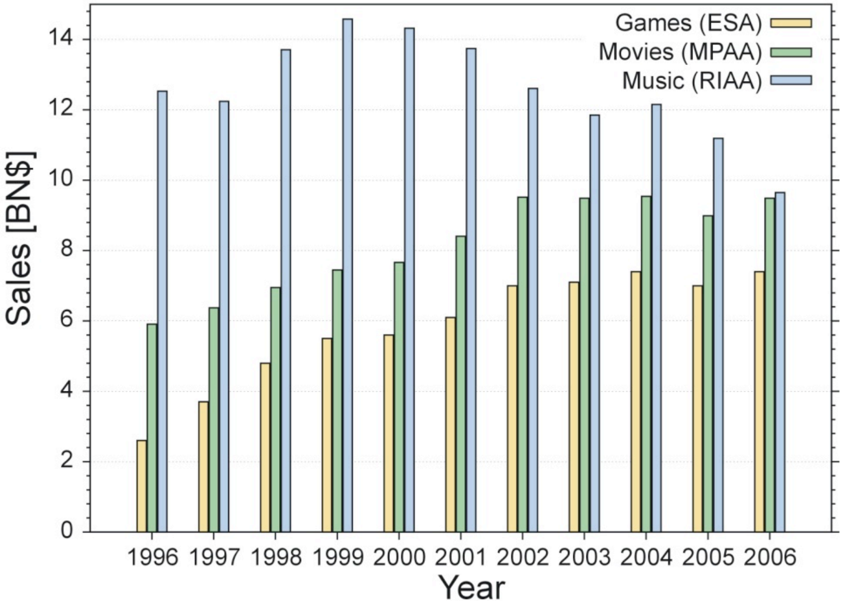
1. **Virtual world**
   explore, do, learn, socialize, compete
   +
2. **Content**
   graphics, maps, puzzles, quests
   +
3. **Game data**
   player stats and relationships

MSGs are a popular, growing market

- 25,000,000+ subscribed players (from 250,000,000+ active)
- Over **10,000 MSGs** in operation
- Subscription market size **$7.5B+/year**, Zynga $600M+/year

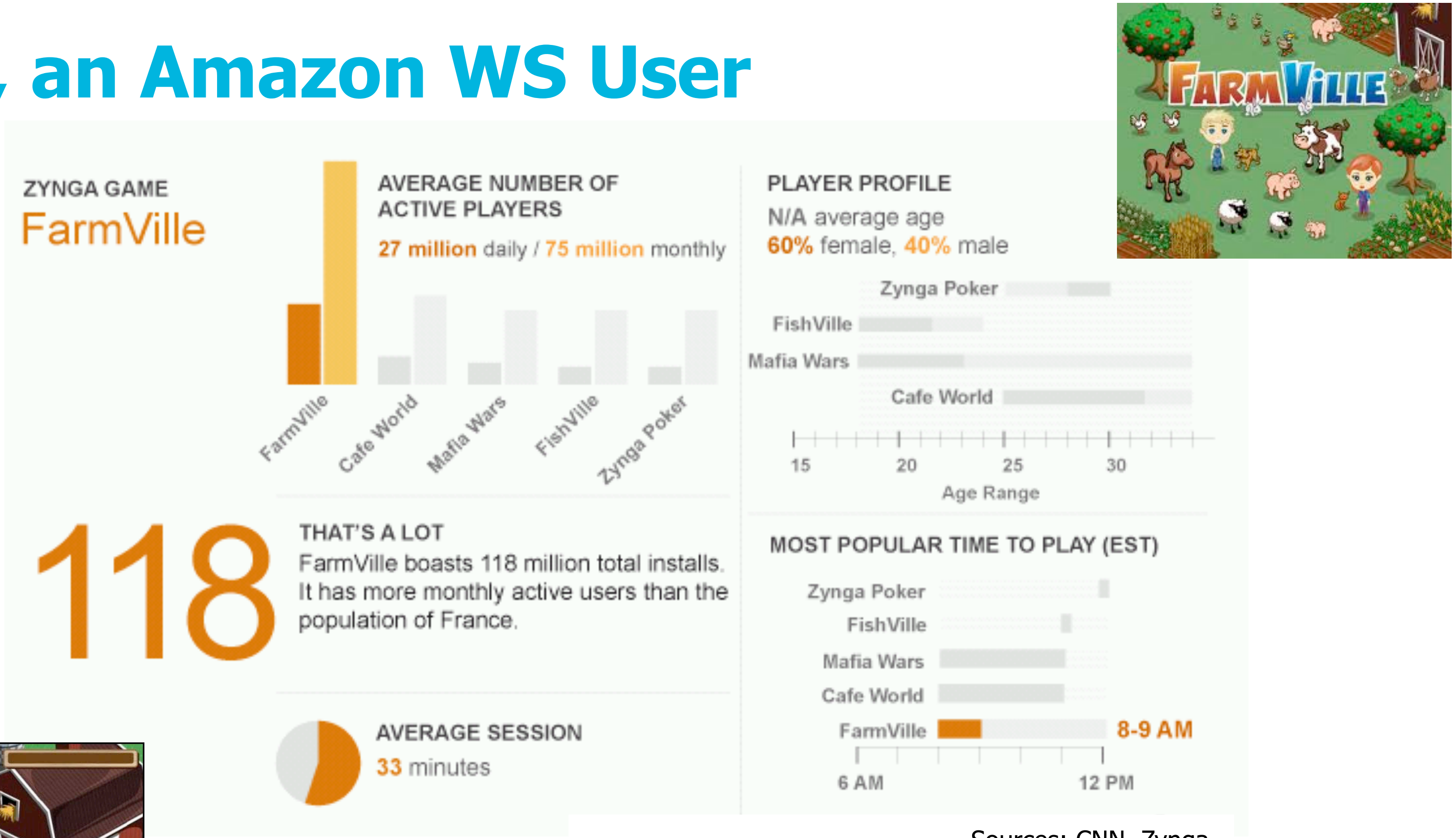


Sources: MMOGChart, own research.



Sources: ESA, MPAA, RIAA.

Zynga, an Amazon WS User



THAT'S A LOT
FarmVille boasts 118 million total installs. It has more monthly active users than the population of France.

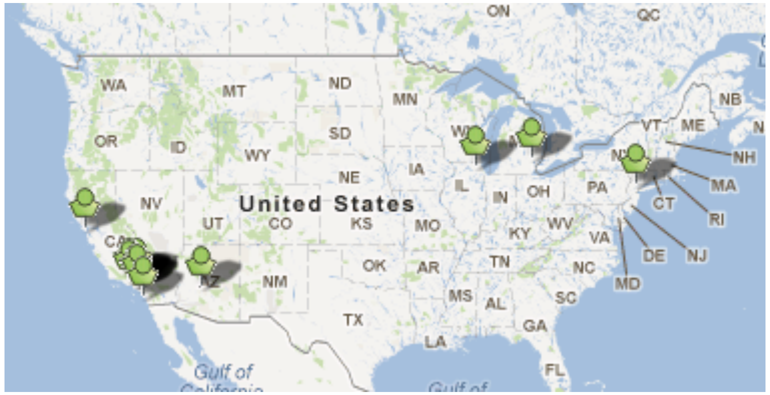
Sources: CNN, Zynga.

Source: InsideSocialGames.com

"Zynga made more than $600M in 2010 from selling in-game virtual goods."
S. Greengard, *CACM*, April 2011

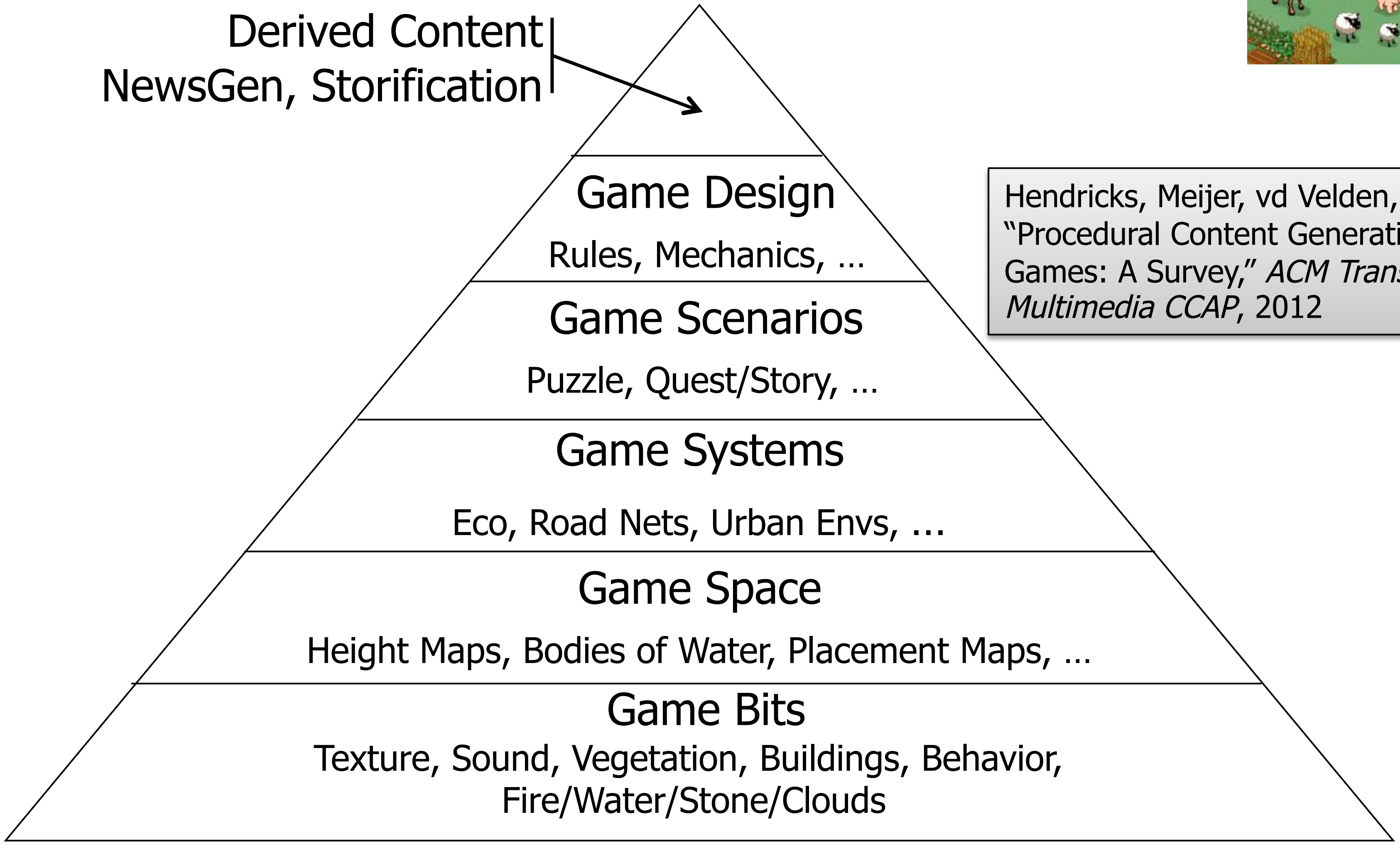
World of Warcraft, a traditional HPC user

- 10 data centers
- 13,250 server blades, 75,000+ cores
- 1.3 PB storage
- 68 sysadmins (1/1,000 cores)



http://www.datacenterknowledge.com/archives/2009/11/25/wows-back-end-10-data-centers-75000-cores/

(Procedural) Game Content (Generation)



Derived Content
NewsGen, Storification

Game Design
Rules, Mechanics, …

Game Scenarios
Puzzle, Quest/Story, …

Game Systems
Eco, Road Nets, Urban Envs, ...

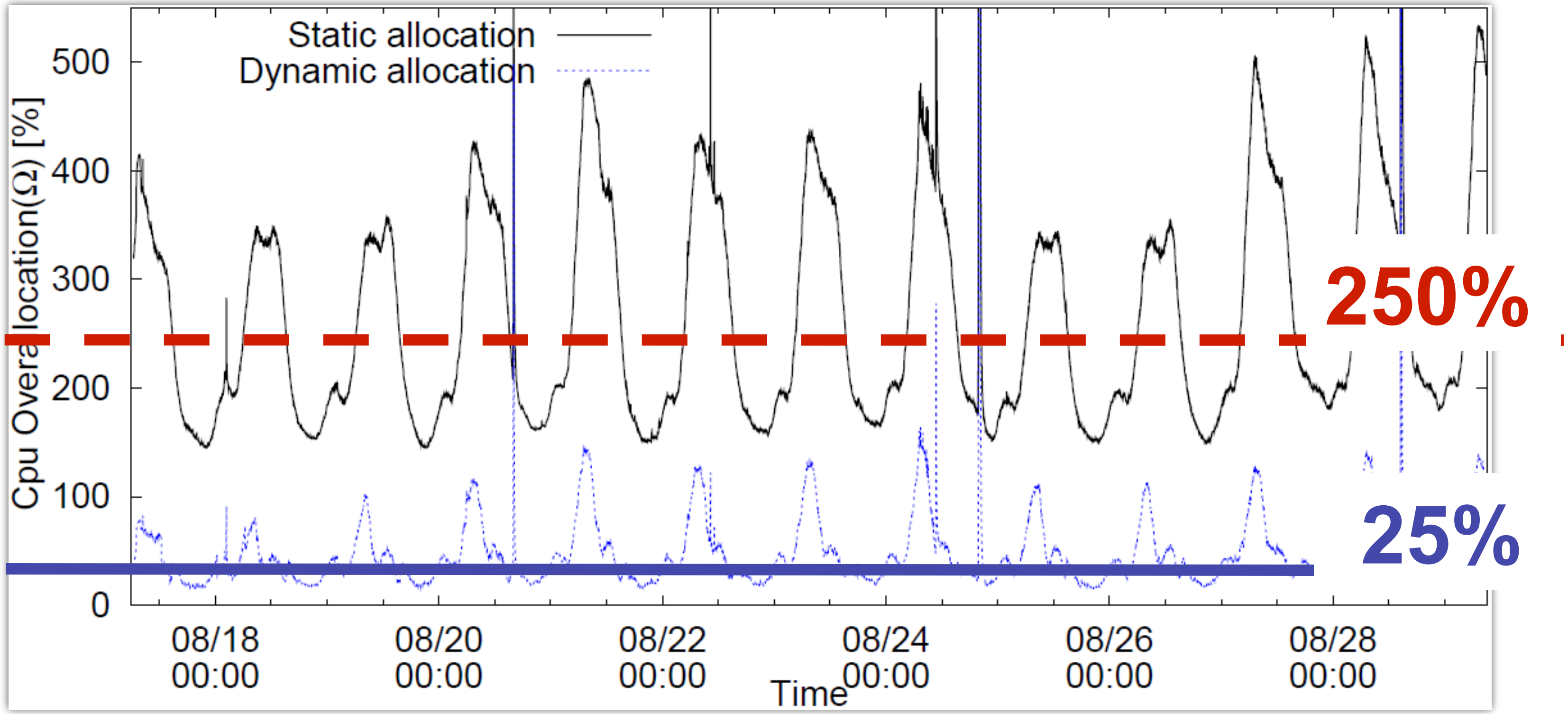
Game Space
Height Maps, Bodies of Water, Placement Maps, …

Game Bits
Texture, Sound, Vegetation, Buildings, Behavior, Fire/Water/Stone/Clouds

Hendricks, Meijer, vd Velden, Iosup, "Procedural Content Generation for Games: A Survey," *ACM Trans. on Multimedia CCAP*, 2012

Resource Provisioning and Allocation

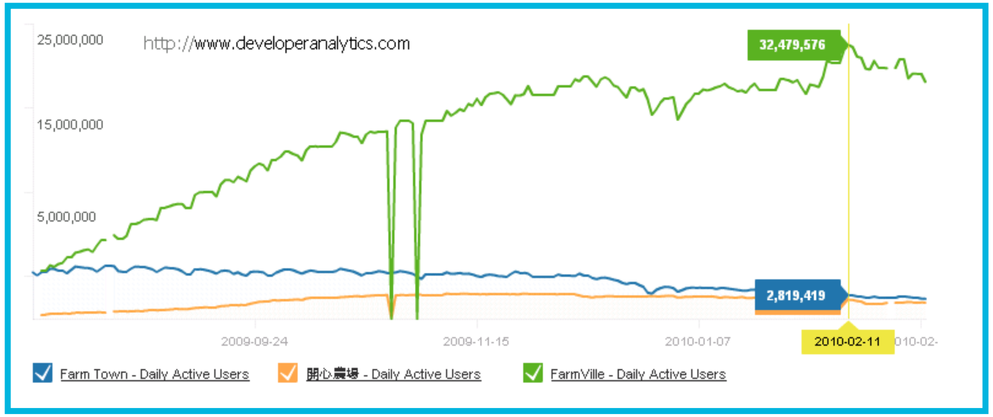
Static vs. **Dynamic** Provisioning



V. Nae, A. Iosup, S. Podlipnig, R. Prodan, D.H.J. Epema, and T. Fahringer, "Efficient Management of Data Center Resources for Massively Multiplayer Online Games," *SuperComputing*, 2008.

@large Research Challenge: V-World Platform for MMOGs

- **Generating content** on time for millions of players
    - player-customized: balanced, diverse, fresh
- **Operational platform scaling** to millions of players
    - 1M in 4 days, 10M in 2 months
- Considerations for both:
    - up-front and operational costs
    - performance, scalability



A. Iosup, "POGGI: Puzzle-Based Online Games on Grid Infrastructures," *Euro-Par 2009* (distinguished paper award)

@large: Social Everything!

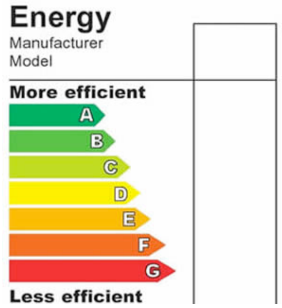
- **Social Network**=undirected graph, **relationship**=edge
- **Community**=sub-graph, density of edges between its nodes higher than density of edges outside sub-graph

Analytics challenge:

Improve the gaming experience

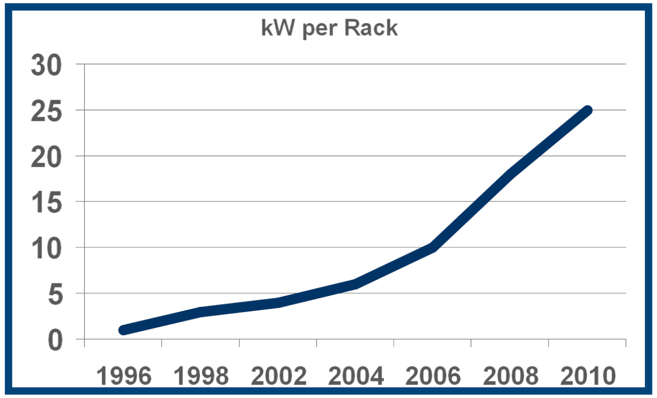
- ranking / rating
- matchmaking / recommendations
- play style / tutoring

A. Iosup, A. Lascateu, N. Tapus, "CAMEO: Enabling Social Networks for Massively Multiplayer Online Games through Continuous Analytics and Cloud Computing," *ACM NetGames*, 2010

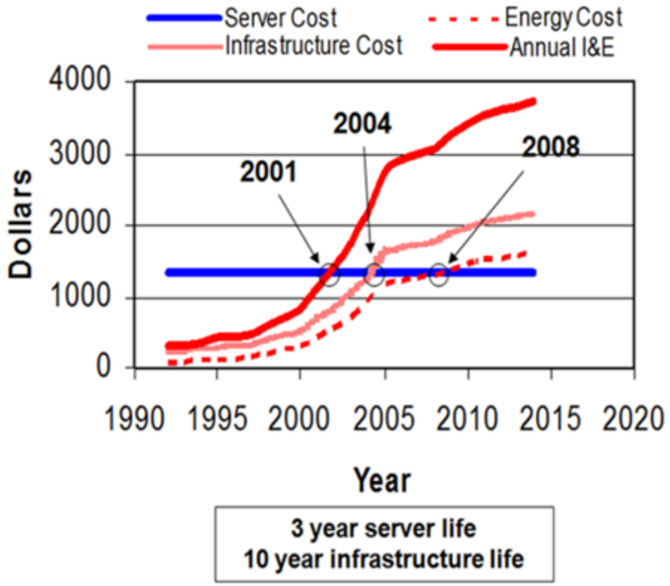


Energy efficiency (1)

data center energy density



power costs vs server costs



Can we exploit **heterogeneity** and **real-time power measurements** for energy-efficient scheduling of MapReduce workloads?

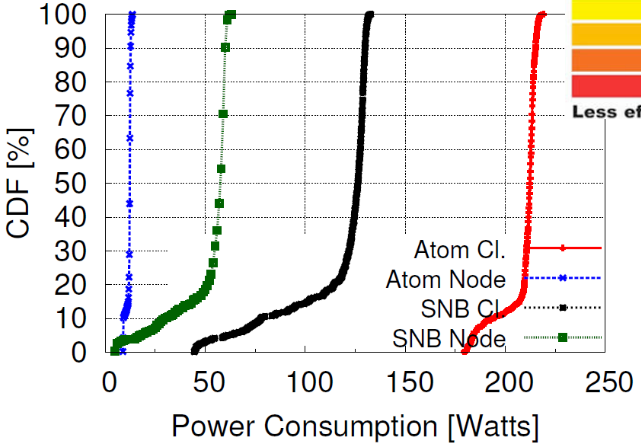
Nezih Yigitbasi, Kushal Datta, Nilesh Jain, and Ted Willke, "Energy Efficient Scheduling of MapReduce Workloads on Heterogeneous Clusters," *2nd International Workshop on Green Computing Middleware* (GCM'2011)

Energy (2): a case for heterogeneity (1)

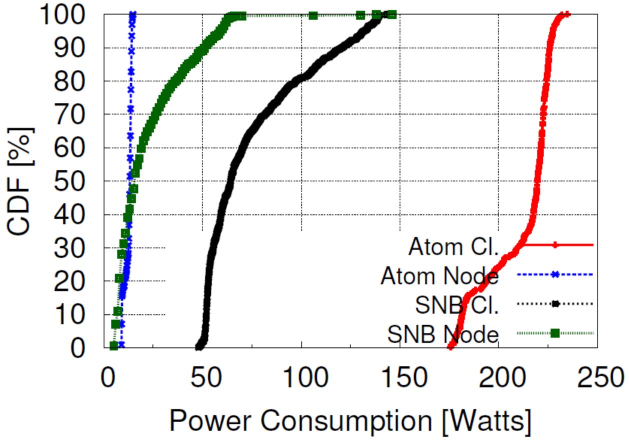
- **Atom** node (wimpy)
  - 2 cores @ 1.66GHz with 4GB memory + SSD
  - narrow dynamic range [9-13W]
  - exploit for I/O bound

- **Sandy Bridge** (SNB) node (brawny)
  - 4 cores @ 3.40GHz with 8GB memory + SSD
  - wide dynamic range [5-150W]
  - Atom:SNB TDP ratio is 1:7

- Atom cluster consumes more power than the SNB cluster
  - ~1.7x for word count, 2.5x for sort and 2.05x for nutch



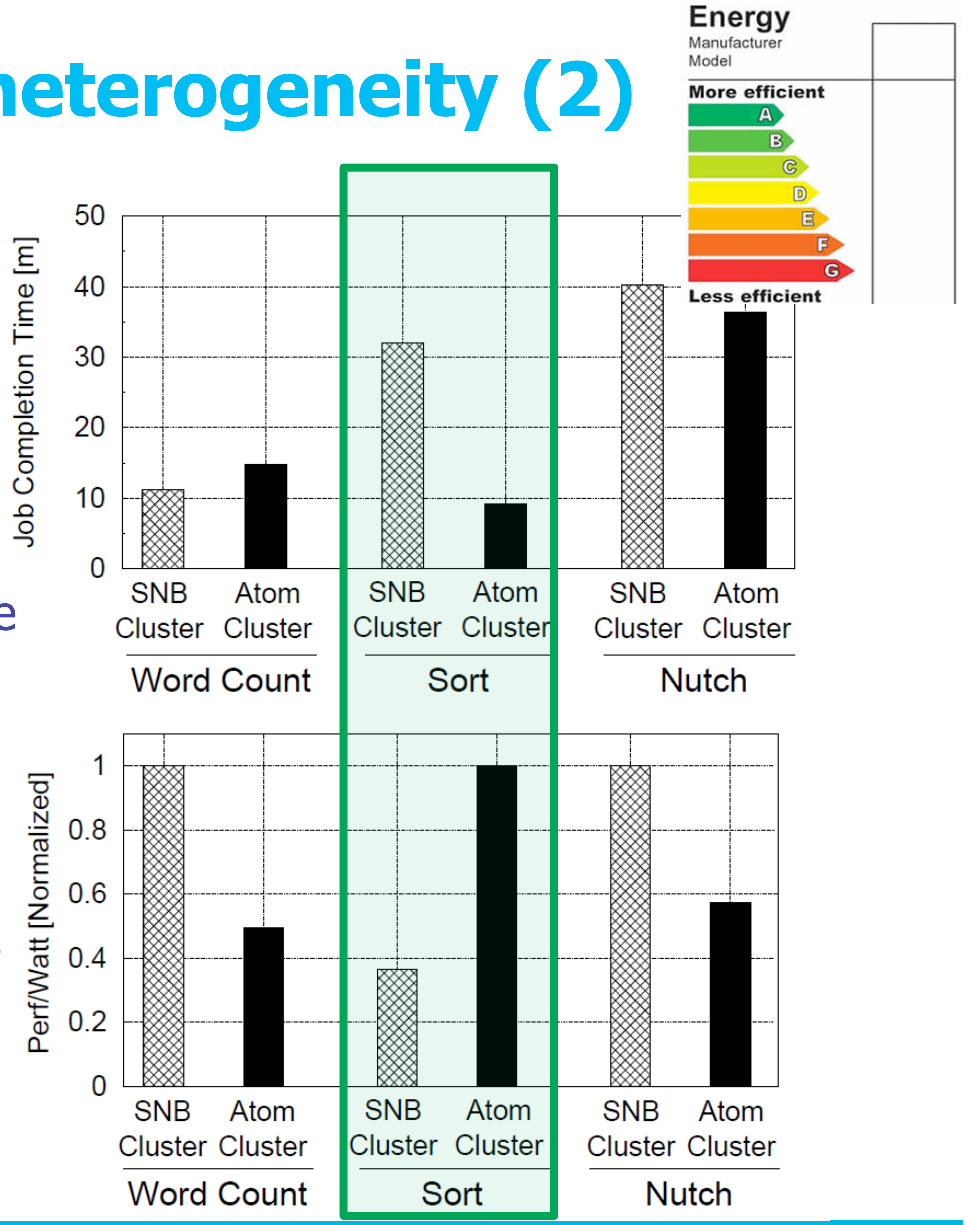
(a) Word count



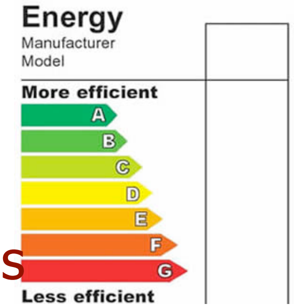
(b) Sort

Energy (3): a case for heterogeneity (2)

- **CPU bound** word count workload
  - Atom has ~1.3x higher completion time
  - SNB has ~2x better energy efficiency
- **I/O bound** sort workload
  - Atom has 3.5x better completion time
  - Atom has 2.5x better energy efficiency
- **Balanced** Nutch workload
  - Atom has slightly better performance but consumes more power
  - SNB has ~1.7x better energy efficiency

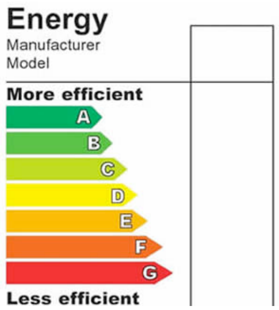


Energy (4): experimental setup

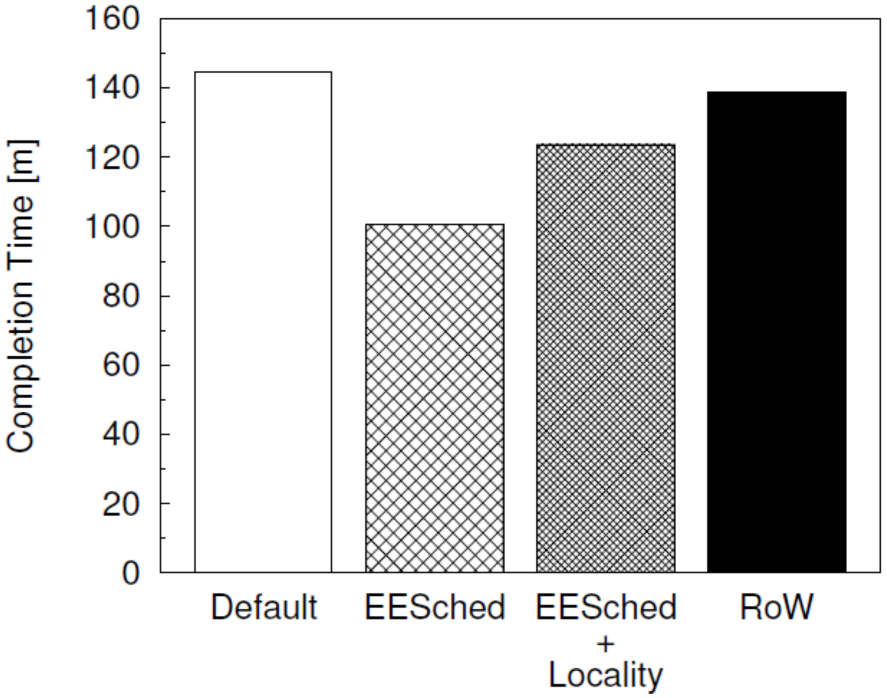


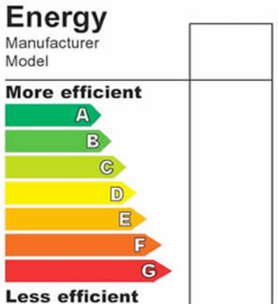
- **Heterogeneous cluster** with 20 Atom nodes and 3 SNB nodes
- **Workload mix** consisting of 25 jobs
  - each job has 15 GB input to process
  - in total, 4,900 map tasks + 800 reduce tasks
- A job can be word count, sort, or nutch
- Job interarrival time follows exp. distribution with a mean of 14 s
  - derived from Facebook Hadoop traces [Zaharia' 10]
- **Scheduling policies**:
  - **EESched**: schedule tasks on most efficient CPU type for the task
  - **EESched+locality**: schedule tasks on a CPU with req. data, then most eff.
  - **RoW**: all reduce tasks on Wimpy (reduce phase is mostly I/O bound)

Energy (5): completion time



- All heuristics **reduce the completion time**
- EESched+Locality worse than EESched
  - HDFS replicates in a random way
  - so a CPU-intensive task may run on a wimpy node
- RoW improvements due to the performance improvements in the reduce tasks



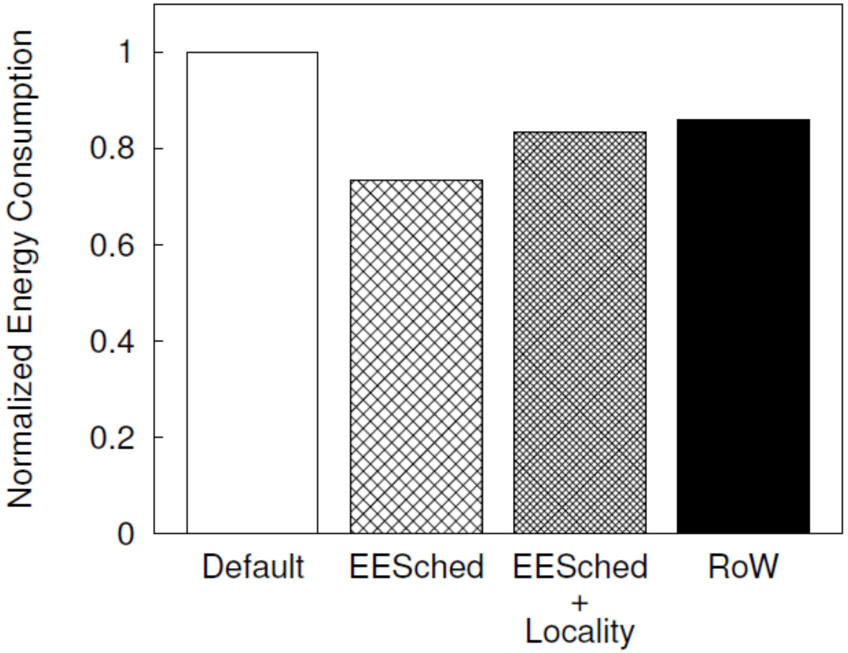


Energy (6): efficiency

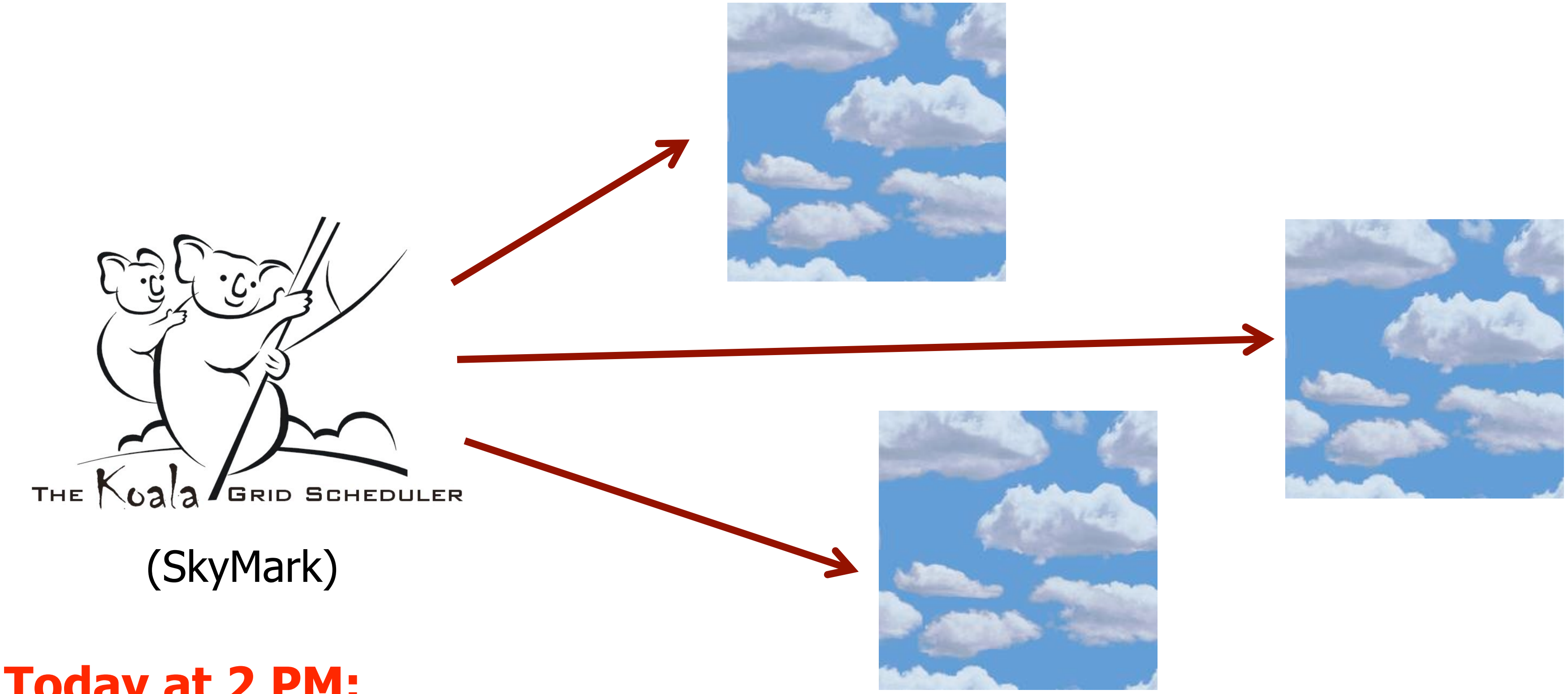
- All heuristics **increase the efficiency**
- EESched+Locality worse than EESched
  - nodes with the input of a task are not necessarily the most energy efficient
- RoW has **17%** better energy efficiency
  - very simple change to the scheduler !
  - worse than the other heuristics since RoW doesn't consider energy efficiency for the map phase

- **Conclusion**:

  Up to 27% better energy efficiency by only modifying the Hadoop scheduler



Cloud resource provisioning and allocation



(SkyMark)

Today at 2 PM:

David Villegas, Athanasios Antoniou, Seyed Masoud Sadjadi and Alexandru Iosup, "An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds", *CCGrid 2012*

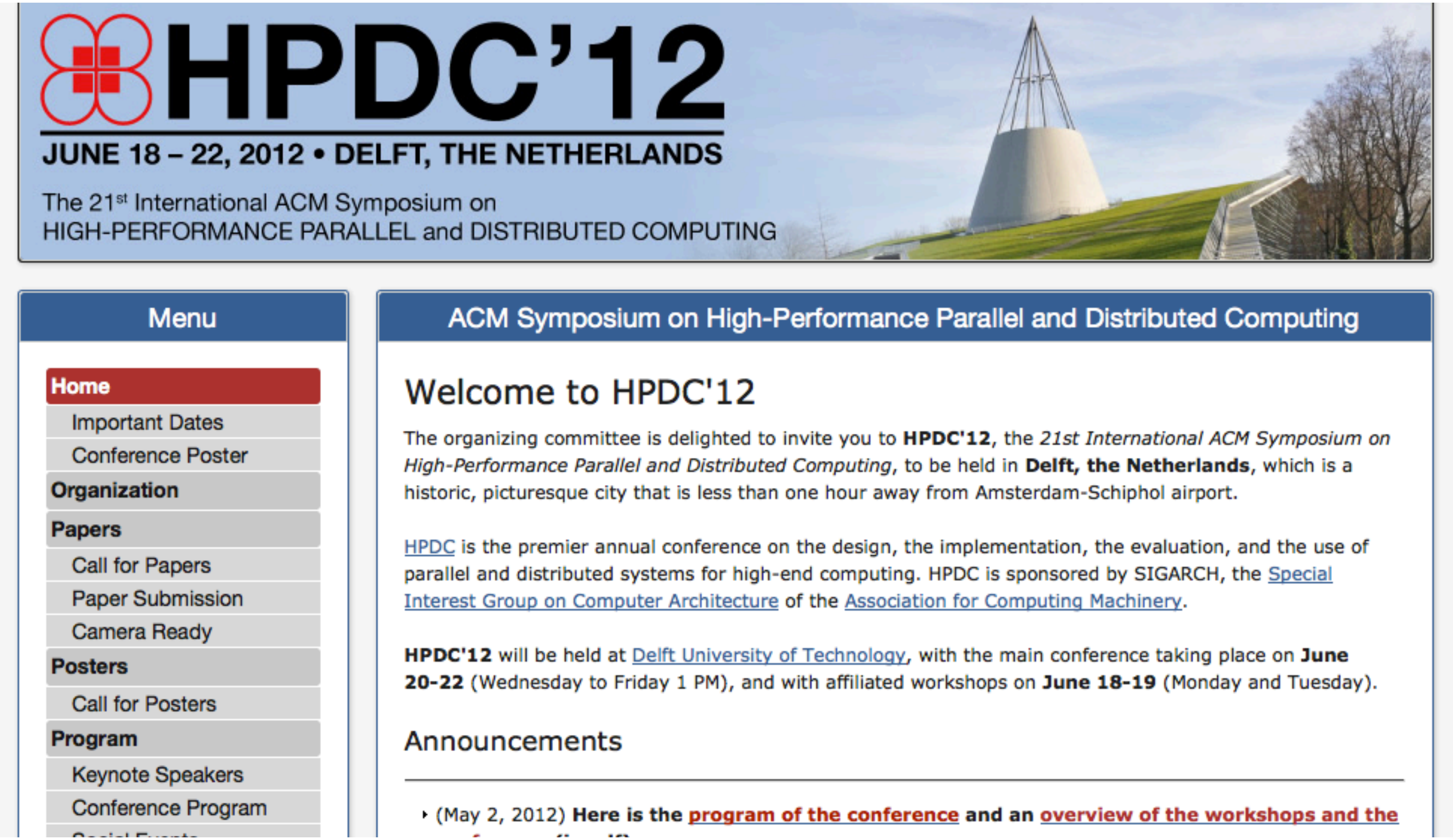
Thanks to

Alexandru Iosup

- Mark van Ameijden (MSc)
- Shanny Anoep (MSc)
- Anasthasios Antoniou (MSc)
- Anca Bucur (PhD)
- Jeremy Buissot (post-doc)
- Catalin Dumitrescu (postdoc)
- Matthieu Gallet (MSc)
- *Bogdan Ghit (PhD student)*
- Bart Grundeken (MSc)
- Alexandru Iosup (PhD, now assist. prof.)
- Mathieu Jan (postdoc)
- Wouter Lammers (MSc)
- Hashim Mohamed (PhD)
- *Thomas de Ruiter (MSc)*
- *Siqi Shen (PhD student)*
- Ozan Sonmez (PhD)
- Corina Stratan (postdoc)
- *Nezih Yigitbasi (PhD student)*

June 18-22, 2012 in Delft

www.hpdc.org/2012



HPDC'12

JUNE 18 – 22, 2012 • DELFT, THE NETHERLANDS

The 21st International ACM Symposium on
HIGH-PERFORMANCE PARALLEL and DISTRIBUTED COMPUTING

ACM Symposium on High-Performance Parallel and Distributed Computing

Welcome to HPDC'12

The organizing committee is delighted to invite you to **HPDC'12**, the *21st International ACM Symposium on High-Performance Parallel and Distributed Computing*, to be held in **Delft, the Netherlands**, which is a historic, picturesque city that is less than one hour away from Amsterdam-Schiphol airport.

HPDC is the premier annual conference on the design, the implementation, the evaluation, and the use of parallel and distributed systems for high-end computing. HPDC is sponsored by SIGARCH, the Special Interest Group on Computer Architecture of the Association for Computing Machinery.

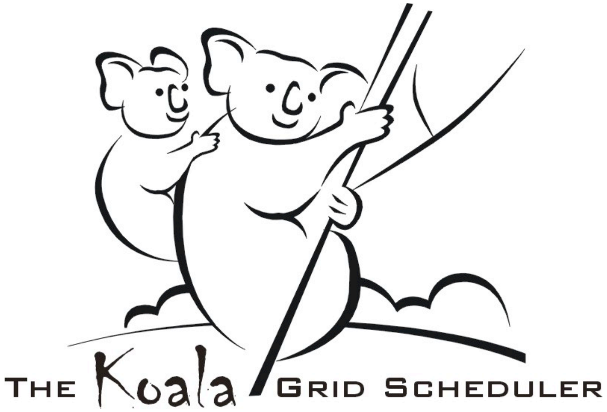
HPDC'12 will be held at Delft University of Technology, with the main conference taking place on **June 20-22** (Wednesday to Friday 1 PM), and with affiliated workshops on **June 18-19** (Monday and Tuesday).

Announcements

- (May 2, 2012) **Here is the program of the conference and an overview of the workshops and the**

More information

- **Publications**
  - see PDS publication database at www.pds.ewi.tudelft.nl/research-publications/publications
- **Home pages**:
  - www.pds.ewi.tudelft.nl/epema
  - www.pds.ewi.tudelft.nl/~iosup
- **Web sites:**
  - KOALA: www.st.ewi.tudelft.nl/koala
  - DAS4: www.cs.vu.nl/das4
  - GUARD-G: guardg.st.ewi.tudelft.nl
  - VL-e: www.vl-e.nl
  - GWA: gwa.ewi. tudelft.nl (grid workload archive)
  - FTA: fta.inria.org (failure trace archive)



CCGrid 2013
MAY 13-16, 2013 • DELFT, THE NETHERLANDS
The 13th IEEE/ACM International Symposium on
Cluster, Cloud and Grid Computing

www.pds.ewi.tudelft.nl/ccgrid2013

Dick Epema
Delft University of Technology
Delft, the Netherlands