

OnVector 2010

The Power of Change!

Cees de Laat

EU

SURFnet

BSIK

NWO

University of Amsterdam

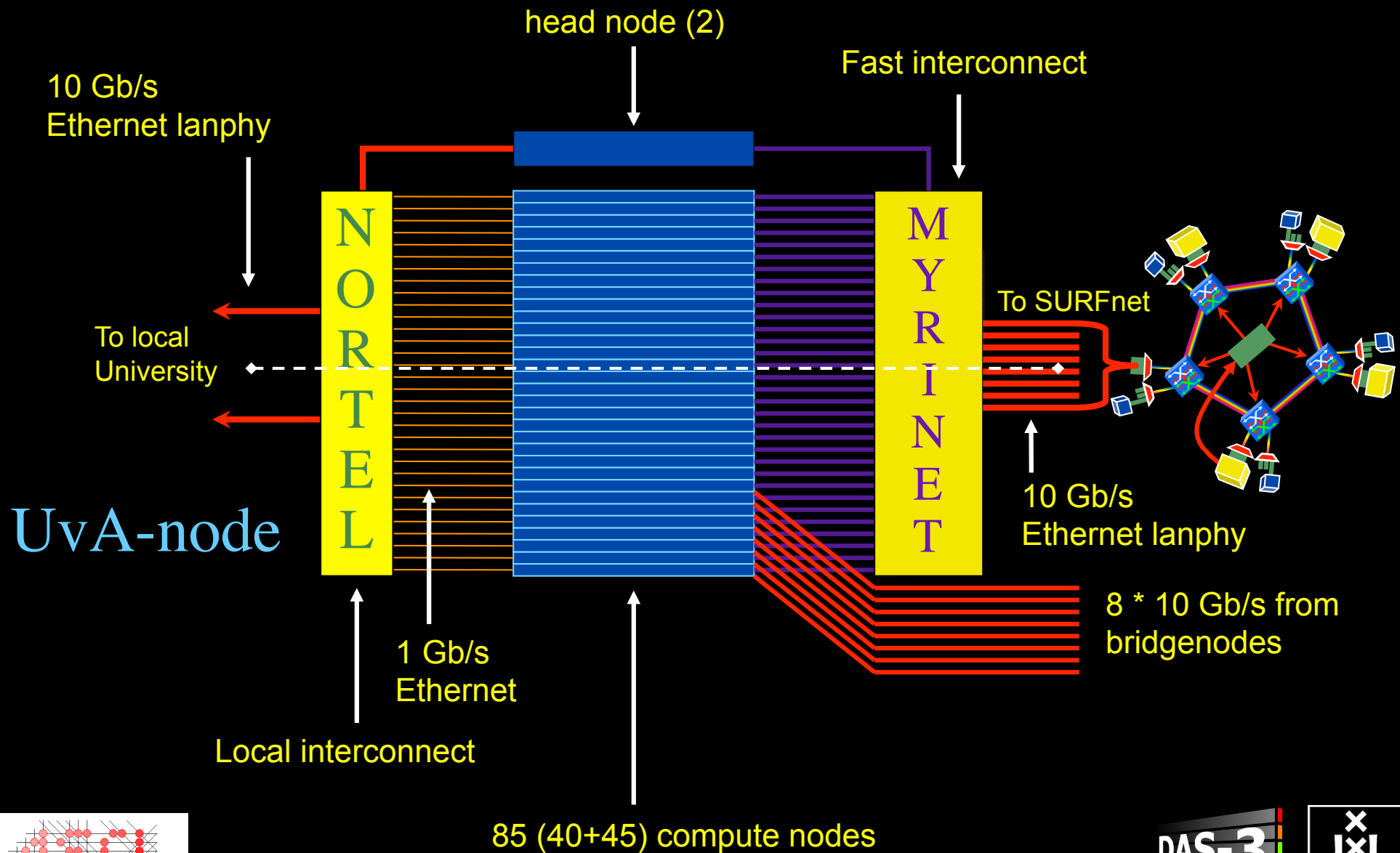
TNO



Themes for next years

- 40 and 100 Gbit/s
- Network modeling and simulation
- **Cross domain Alien Light switching**
- **GreenLight - GreenSonar**
- **Network and infrastructure descriptions & WEB2.0**
- **Reasoning about services**
- Cloud Data - Computing
- Web Services based Authorization
- Network Services Interface (N-S and E-W)
- Fault tolerance, Fault isolation, **Monitoring**
- eScience integrated services
- Data and Media specific services
- **→ Smart e-Infrastructure**

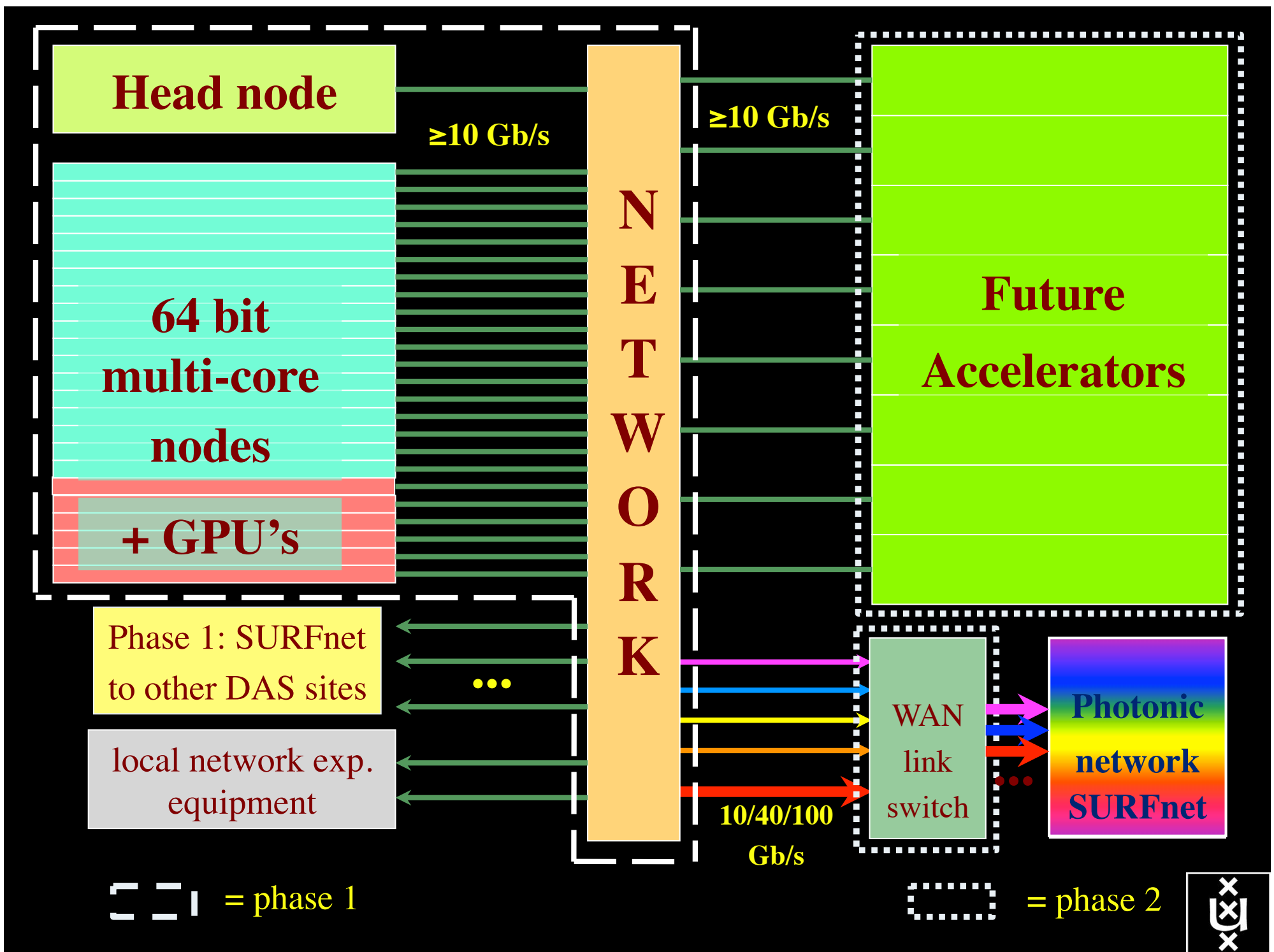
DAS-3 Cluster Architecture



Power is a big issue

- UvA cluster uses (max) 30 kWh
- 1 kWh ~ 0.1 €
- per year -> 26 k€/y
- add cooling 50% -> 39 k€/y
- Emergency power system -> 60 k€/y
- over 4 year = 240 kEuro for a 500 kEuro set.
- per rack 15 kWh is now normal
- **YOU BURN HALF THE CLUSTER OVER ITS LIFETIME!**





u
s
e
r
s

A. Lightweight users, browsing, mailing, home use

Need full Internet routing, one to all

B. Business/grid applications, multicast, streaming, VO's, mostly LAN

Need VPN services and full Internet routing, several to several + uplink to all

C. E-Science applications, distributed data processing, all sorts of grids

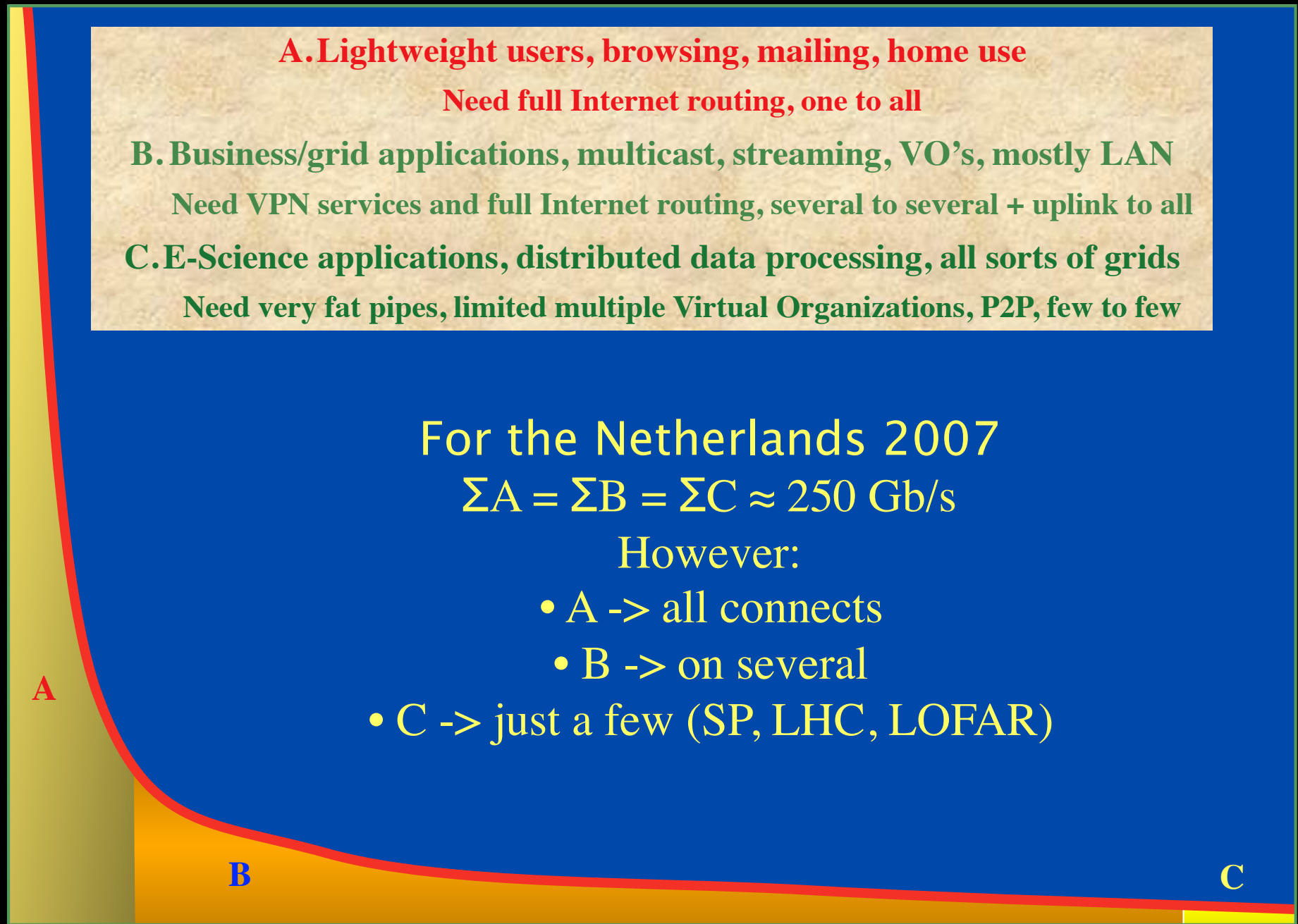
Need very fat pipes, limited multiple Virtual Organizations, P2P, few to few

For the Netherlands 2007

$$\Sigma A = \Sigma B = \Sigma C \approx 250 \text{ Gb/s}$$

However:

- A -> all connects
- B -> on several
- C -> just a few (SP, LHC, LOFAR)



ADSL (12 Mbit/s)

GigE

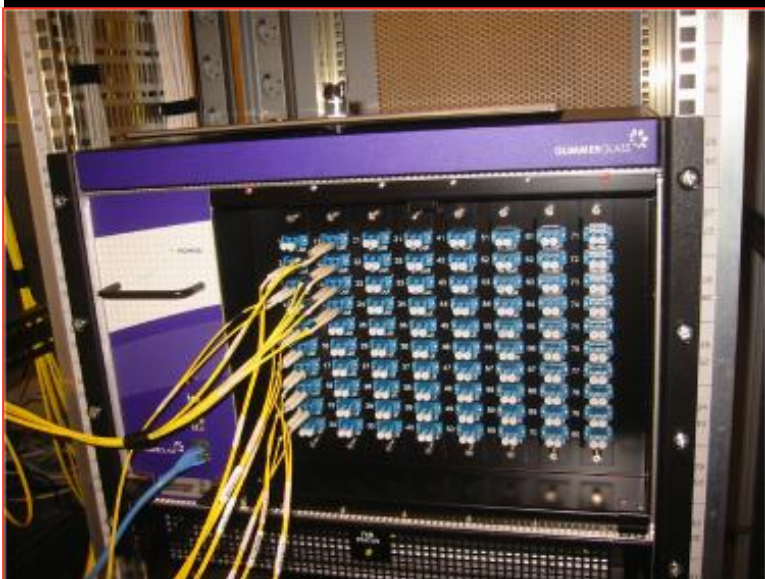
BW requirements



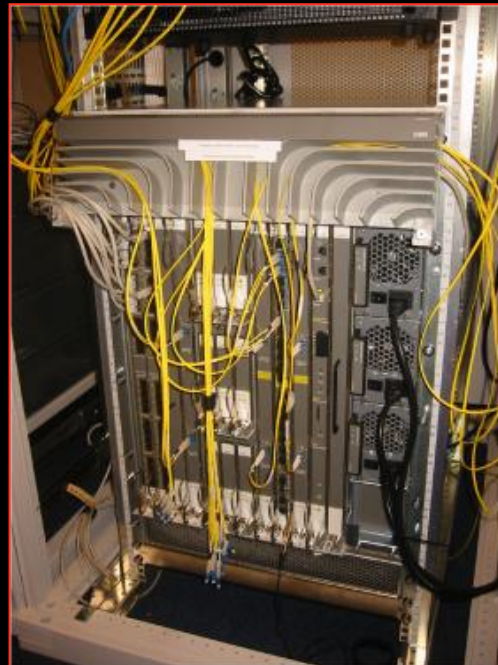
Towards Hybrid Networking!

- Costs of photonic equipment 10% of switching 10 % of full routing
 - for same throughput!
 - Photonic vs Optical (optical used for SONET, etc, 10-50 k\$/port)
 - DWDM lasers for long reach expensive, 10-50 k\$
- Bottom line: look for a hybrid architecture which serves all classes in a cost effective way
 - map A -> L3 , B -> L2 , C -> L1 and L2
- Give each packet in the network the service it needs, but no more !

L1 \approx 2-3 k\$/port



L2 \approx 5-8 k\$/port

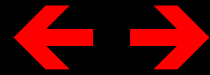


L3 \approx 75+ k\$/port



Hybrid computing

Routers



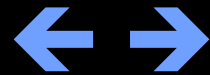
Supercomputers

Ethernet switches



Grid & Cloud

Photonic transport



GPU's

What matters:

Energy consumption/multiplication

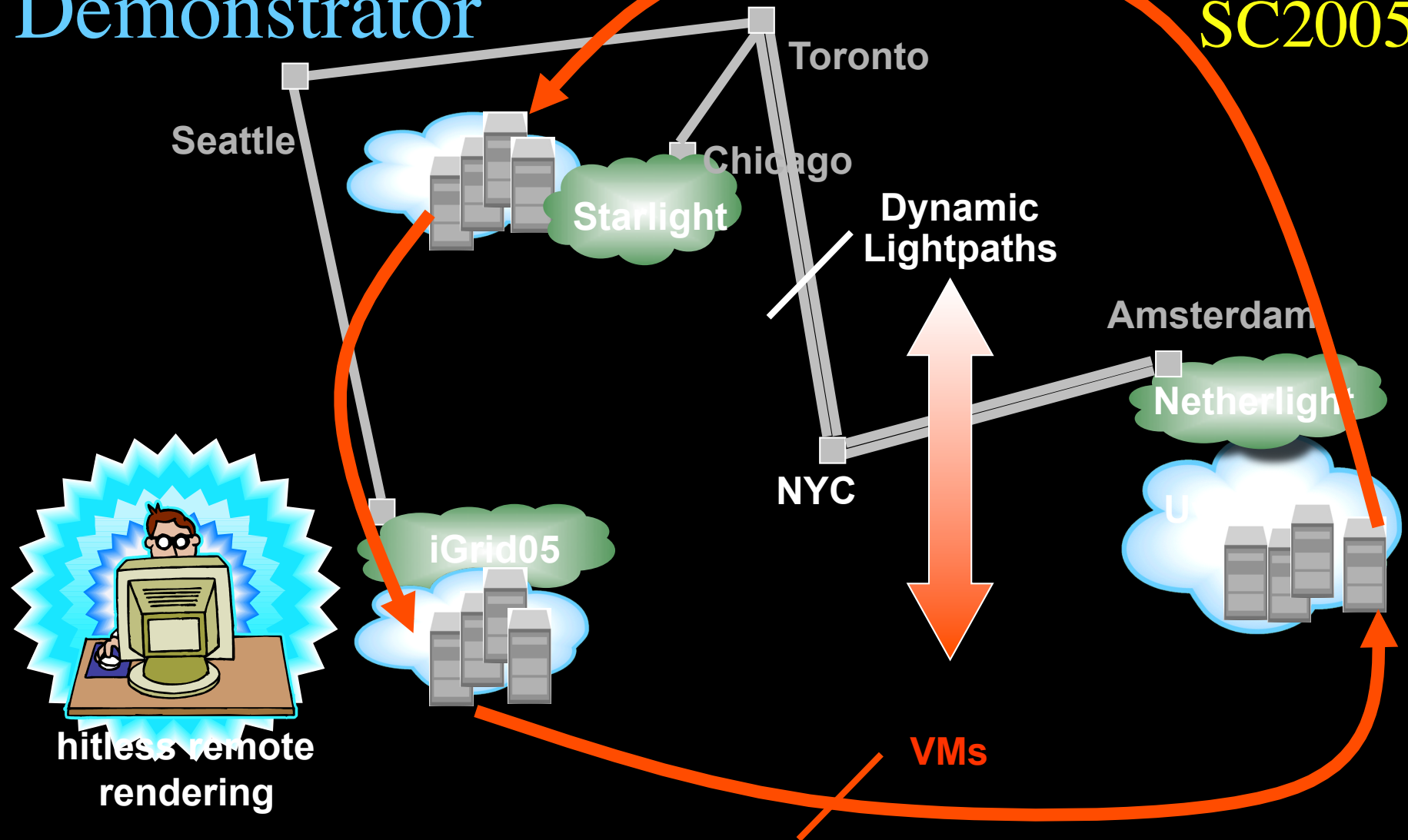
Energy consumption/bit transported



The VM Turntable Demonstrator

iGrid2005

SC2005



The VMs that are live-migrated run an iterative search-refine-search workflow against data stored in different databases at the various locations. A user in San Diego gets hitless rendering of search progress as VMs spin around

CosmoGrid

Supercomputing Grid across Continents and Oceans

And yes, it works!

Application

We originally developed MPWide to manage the long-distance message passing in the CosmoGrid[†] project. This is a large-scale cosmological project whose primary goal is to perform a dark matter simulation using supercomputers on two continents.

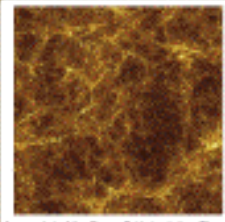
In this simulation, we use the cosmological Λ Cold Dark Matter model[‡] to simulate the dark matter particles using a parallel tree/particle-mesh N-body integrator, TreePM[§]. This requires relatively little communication between different sites after each timestep. This integrator calculates the dynamical evolution of 2048³ (8.5 billion) particles. More information about the parameters used and the scientific rationale can be found in [¶].

The integrator can be run as a single MPI application, or as two separately launched MPI applications on different supercomputers.

[¶] Portegies Zwart et al., 2009, IEEE Computer (submitted)

[§] Gahn, 1981: Physical Review D

[‡] Yoshikawa and Fukushige, 2005: PASJ

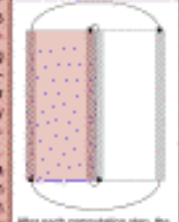


A snapshot of the CosmoGrid simulation. The bright dense areas form a cosmic web structure.

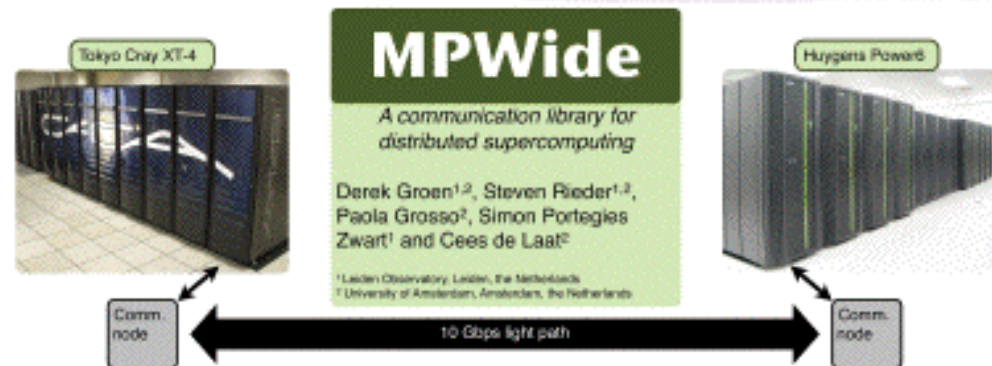
Motivation

We use MPWide to manage the wide area communications in the CosmoGrid project, where cosmological N-body simulations run on grids of supercomputers connected by high performance optical networks. To take full advantage of the network light paths in CosmoGrid, we need a message passing library that supports the ability to use customized communication settings (e.g. custom number of streams, window sizes) for individual network links among the sites. The supercomputers see use vary both in hardware architectures and software setup.

Many supercomputers have a recommended MPI implementation which has been optimized for the network architecture of that particular machine. Installing and optimizing a homogeneous MPI implementation on multiple supercomputer platforms is a task that may be politically difficult to initiate, and requires considerable effort and man hours to complete. This has led us to develop MPWide, a light-weight communication library which connects two applications, each of them running with the locally recommended MPI implementation.



After each computation step, the data in grey regions is transferred to the other supercomputer.



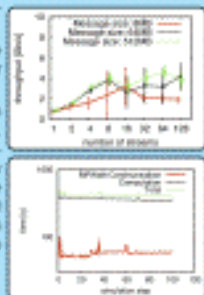
Benchmarks

We measured the performance of MPWide between two nodes on different supercomputers, one located in The Netherlands, the other in Finland. These supercomputers are connected with a 10 Gbps interface. The round trip time for this network is 37.6 ms.

Each test consists of 100 two-way message exchanges, where we record the average throughput and the standard error. We performed the tests over a shared network with frequent background traffic.

Our tests show increased performance when using more streams, especially for larger message sizes.

We also tested MPWide in a production environment, during a CosmoGrid run. In this run, we used the Huygens supercomputer in Amsterdam and the Cray supercomputer in Tokyo. In this run, the calculation time dominated the overall performance, with the communication time constituting about one eighth of the total execution time.



Related work and future

The MPI implementation most closely related to our work is the PACX-MPI[¶] implementation. Like MPWide, this implementation connects different machines, while making use of the vendor MPI library on the system. The main difference between PACX-MPI and MPWide lies in the fact that MPWide supports a de-centralized startup, where PACX-MPI does not. For CosmoGrid, support for this is required, as it is not possible to start the simulation on all supercomputers from one site.

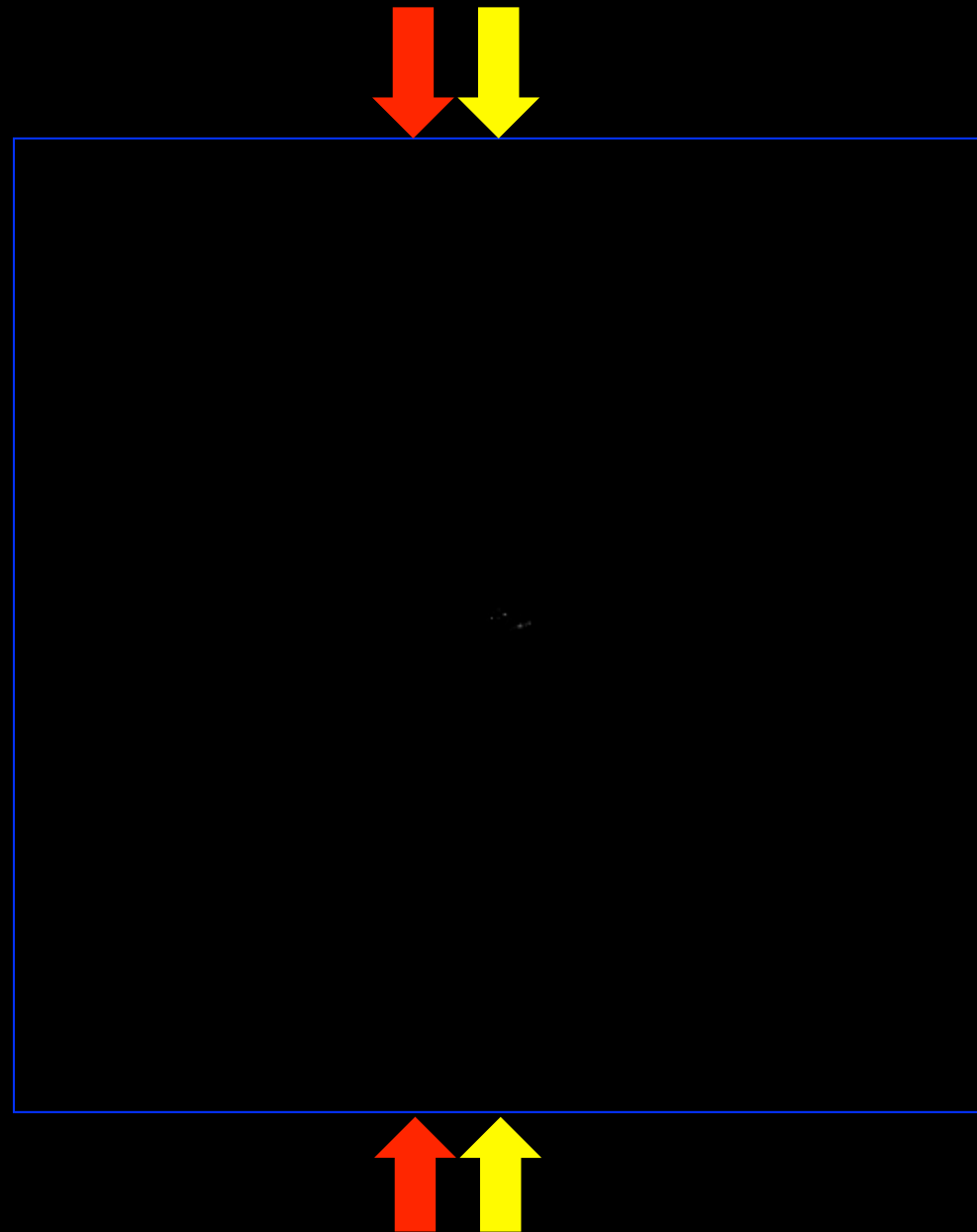
Other implementations of MPI, like Open MPI and MPICH-G2, differ further from MPWide, and do not support manual specification of the network topology, required by CosmoGrid.

In the near future, we will expand the CosmoGrid simulation to run on four supercomputer sites, and we will implement support for this in MPWide.



[¶] <http://www.his.de/organization/axsm/research/pacx-mpi/>

Auto-balancing Supers



Interactive programmable networks



SCARIE Programmable networks to distribute work

Network Control in Distributed Computing



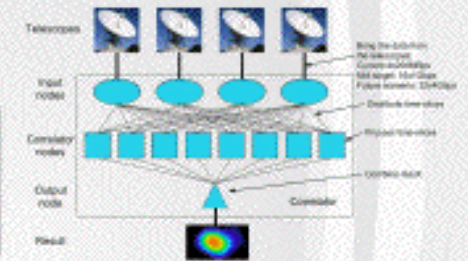
Motivation: when over-dimension is unfeasible

- Large-scale observation systems monitor environmental objects such as dikes in order to prevent disasters, or watch radio wave emissions from stars.
- Large-scale observation systems are dynamic in their resource demands.
- Large-scale observation systems need distributed computing where the available resources are used in an optimal way. Hence, infrastructure topology does matter!
- Distributed applications need specific network services and the ability to optimize their needs.
- Distributed computing platforms, such as Grids or Clouds need application support for network service development, deployment and management.



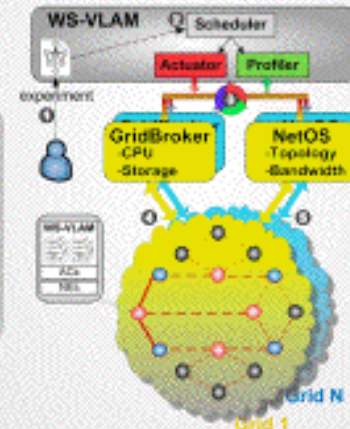
Applications that require specific network services that current distributed systems lack to deliver:

- **Udijk**: large-scale sensor networks collect enormous amount of environmental data such as dikes and push the data into forecast models in order to predict dangerous events.
 - **SCARIE**: a Grid-based software correlator for radio-telescope images requires high-throughput communication, but with specific services such as soft real-time or combant throughput.
- Network services can be part of applications or stand-alone distributed programs.

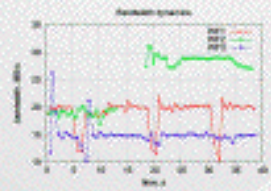
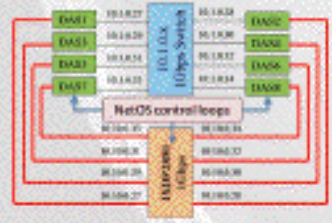


WS-VLAM - workflow execution environment coordinates the execution of distributed Apps

- 1 - User deploys an experiment, application & basic infrastructure requirements.
- 2 - WS-VLAM maps the experiment using Actuator onto available distributed resources as defined by Profiler.
- 3 - Control loops may occur in which WS-VLAM is a controller to adjust the resources such as to solve the applications demands regardless of the environment changes.



- Broker manages the computational resources.
 - NetOS programs the networking infrastructure of distributed system.
- Each node:
- supports the applications running under WS-VLAM supervision
 - provides the application-specific network services through application components & CA's supported by network services NS's



A network showing a distributed system, in which nodes are interconnected through 2 networks, as follows:

- a default network uses a shared 1Gbps gigabit switch
- a second network uses a network processor unit programmed to route IP packets at 10Gbps, 100.

WS-VLAM management starts applications and sets up the paths one by one on the default network (10.1.0.x).

When measured network performance (throughput) decreased below an application threshold, WS-VLAM starts "offloading" the paths from 10.1.0.x network onto 10.10.0.x network.

Management of the programmable network services in a distributed computing needs to a dedicated queuing system for network resources.

¹Mihai Lucian Cristea, ^{1,2}Rudolf Strijkers, ^{1,4}Vladimir Korkhov, ^{1,4}Adam Belloun, ³Mark Kettinis, ³Aard Kotteperu, ¹Coen de Laat, ^{1,2}Robert Meijer

Path finding in multi-domain multi-layer networks

A new approach based on declarative
programming

P. Grosso, A. Taal, L. Xu, J. v/d Ham, C. de Laat



Multi layer multi domain networks

The networks for e-Science where applications use dedicated optical circuits.

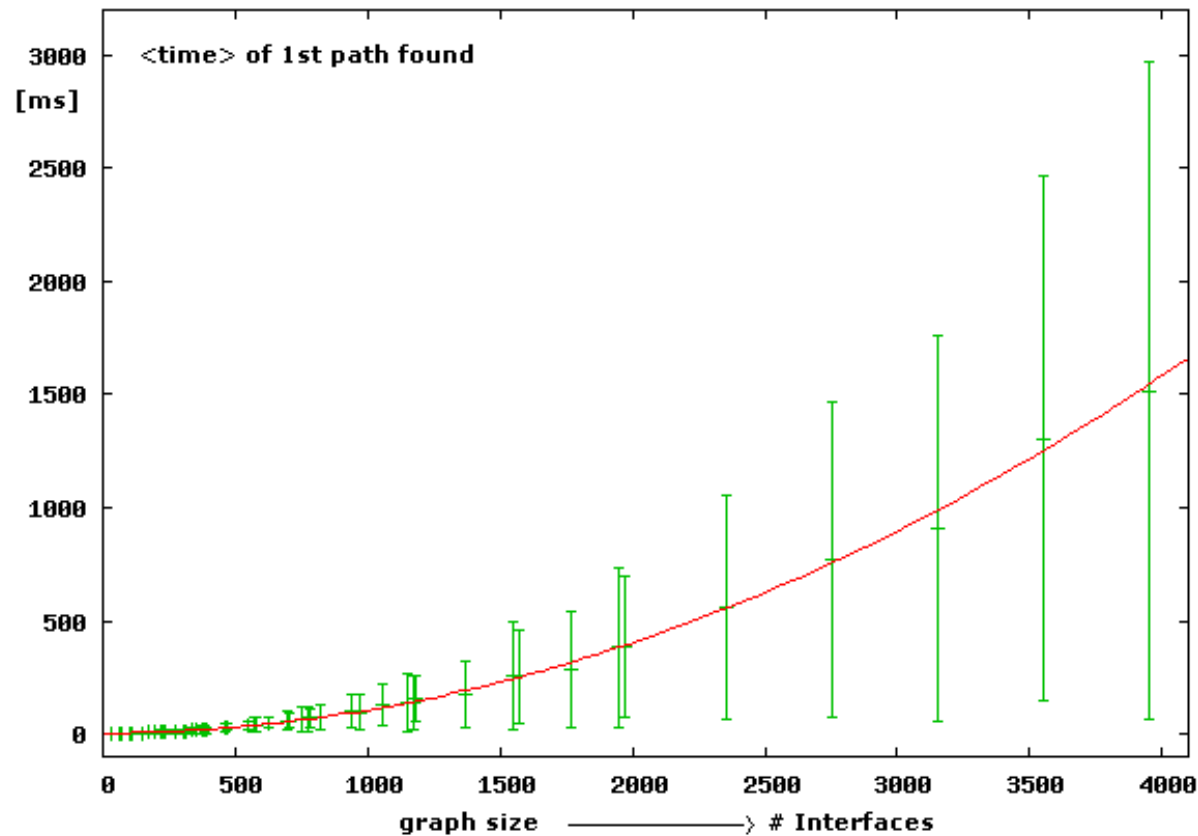
Is declarative programming more suitable to find paths in multi-domain multi-layer networks? Especially in presence of constraints and complex requests?

Our approach:

1. We generate BA network graphs with a varying number of domains and nodes. Barabasi-Albert scale free graphs are a good representation of these networks.
2. We represent the graphs in NDL – Network Description Language, the RDF schemas.
3. We load the RDF files in Prolog and Python programs
4. We perform a modified DFS –Depth First Search- algorithm to find paths.



Single layer networks: results

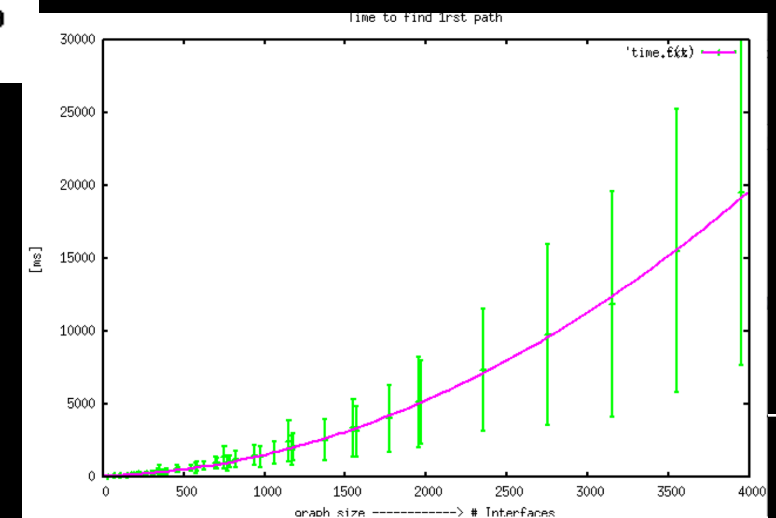


- Number of interfaces,
- given N nodes per domain D
- $4*(D-2) + D*4*(N-2)$ for $D > 2$

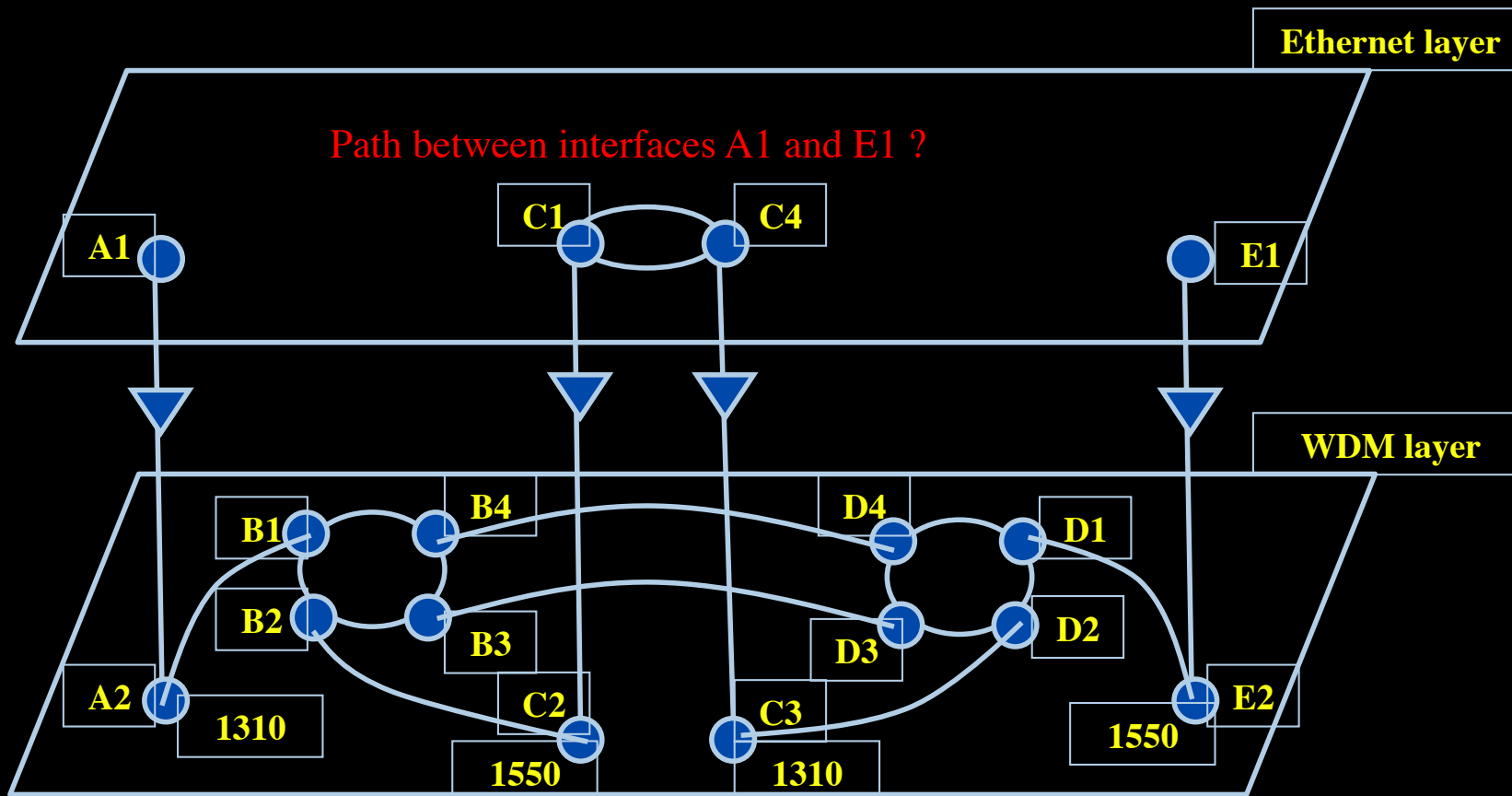
Pynt-based DFS

Prolog DFS

- Prolog time to find first path shorter than Python time.
- We observe a quadratic dependence.
- Length of paths found comparable.



Multi-layer network



Prolog rule:

`linkedto(Intf1, Intf2, CurrWav):-`

`rdf_db:rdf(Intf1, ndl:'layer', Layer),`

`Layer == 'wdm#LambdaNetworkElement',`

`rdf_db:rdf(Intf1, ndl:'linkedTo', Intf2),`

`rdf_db:rdf(Intf2, wdm:'wavelength', W2),`

`compatible_wavelengths(CurrWav, W2).`

`%-- is there a link between Intf1 and Intf2 for wavelength CurrWav ?`

`%-- get layer of interface Intf1 → Layer`

`%-- are we at the WDM-layer ?`

`%-- is Intf1 linked to Intf2 in the RDF file?`

`%-- get wavelength of Intf2 → W2`

`%-- is CurrWav compatible with W2 ?`

`linkedto(B4, D4, CurrWav)` is true for any value of `CurrWav`

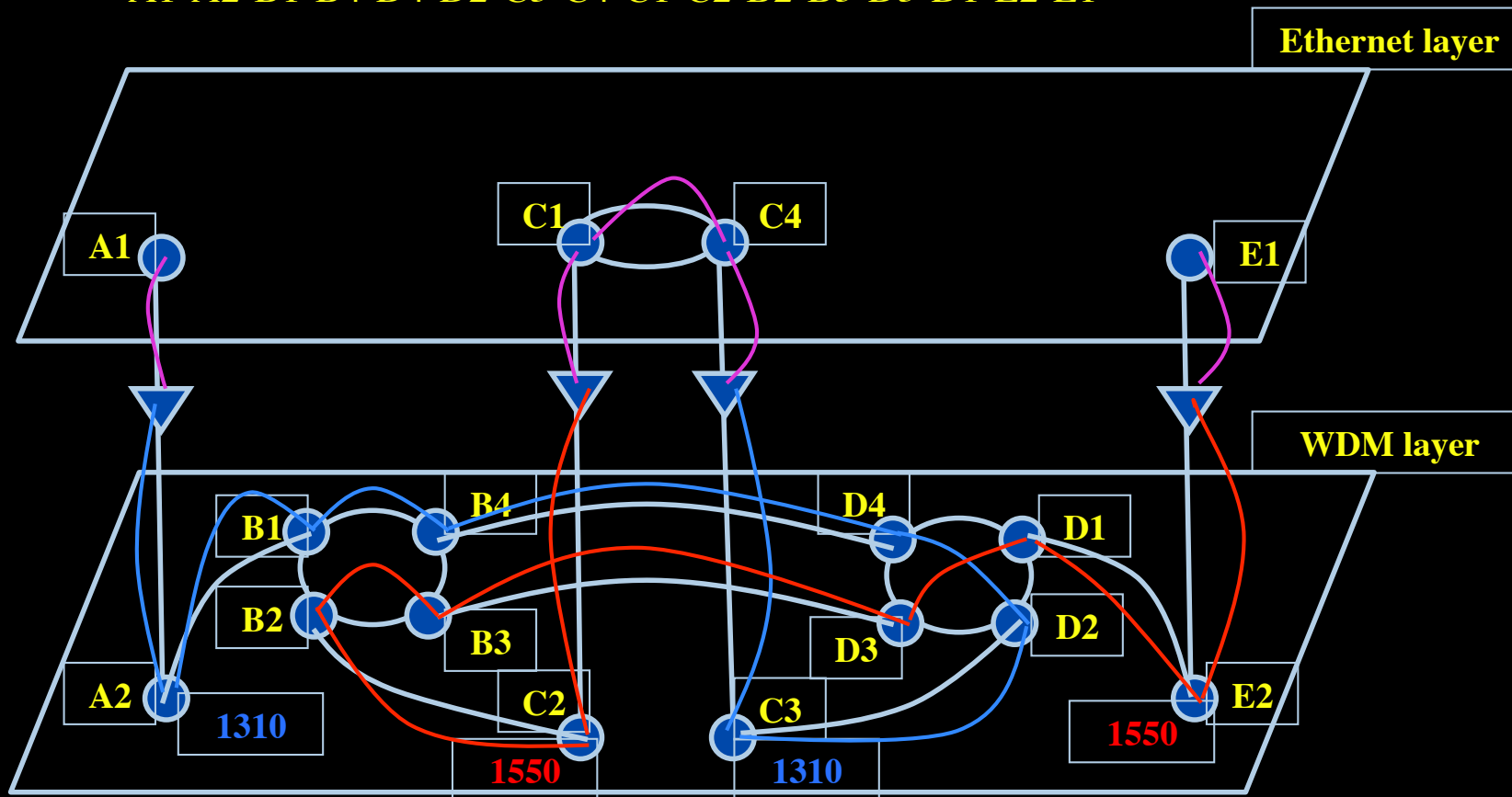
`linkedto(D2, C3, CurrWav)` is true if `CurrWav == 1310`



Multi-layer

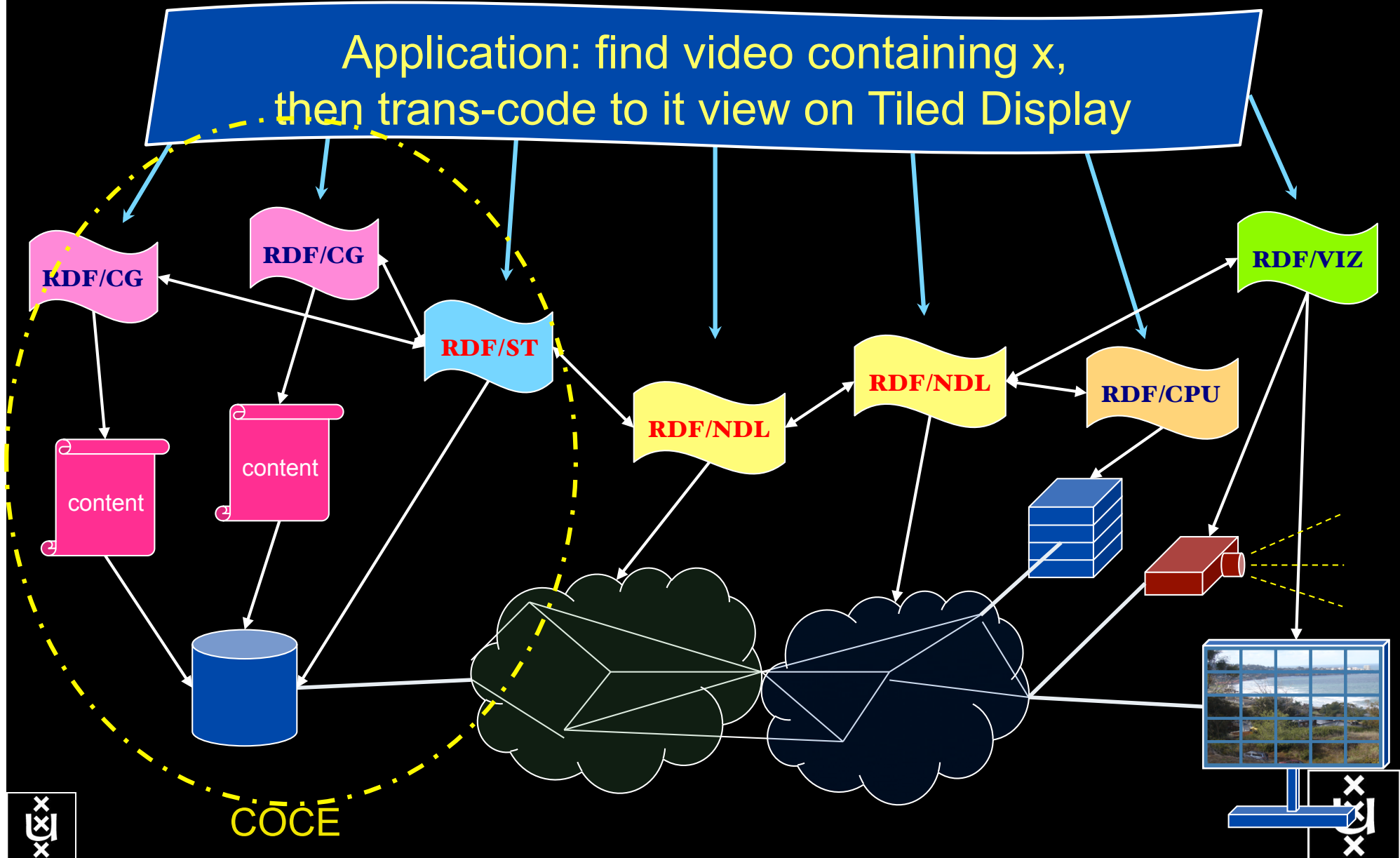
Path between interfaces A1 and E1:

A1-A2-B1-B4-D4-D2-C3-C4-C1-C2-B2-B3-D3-D1-E2-E1



RDF describing Infrastructure “I want”

Application: find video containing x,
then trans-code to it view on Tiled Display



Applications and Networks become aware of each other!

CineGrid Description Language

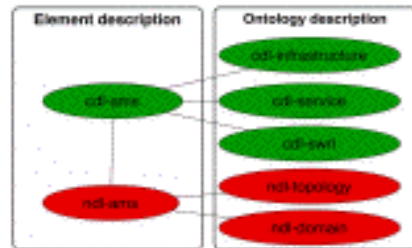
CineGrid is an initiative to facilitate the exchange, storage and display of high-quality digital media.

The CineGrid Description Language (CDL) describes CineGrid resources. Streaming, display and storage components are organized in a hierarchical way.

CDL has bindings to the NDL ontology that enables descriptions of network components and their interconnections.

With CDL we can reason on the CineGrid infrastructure and its services.

UML representation of CDL



SQWRL is used to query the Ontology.



```
cdl:hasElements(?node1, ?host1) ^
ndl-topo:hasInterface(?host1, ?if1) ^ ndl-
topo:connectedTo(?if1, ?if2) ^
ndl-topo:hasInterface(?host2, ?if2) ^
cdl:hasElements(?node2, ?host2) ->
sparql:select(?node1, ?node2)
```



CDL links to NDL using the **owl:SameAs** property. CDL defines the services, NDL the network interfaces and links. The combination of the two ontologies identifies the host pairs that support matching services via existing network connections.



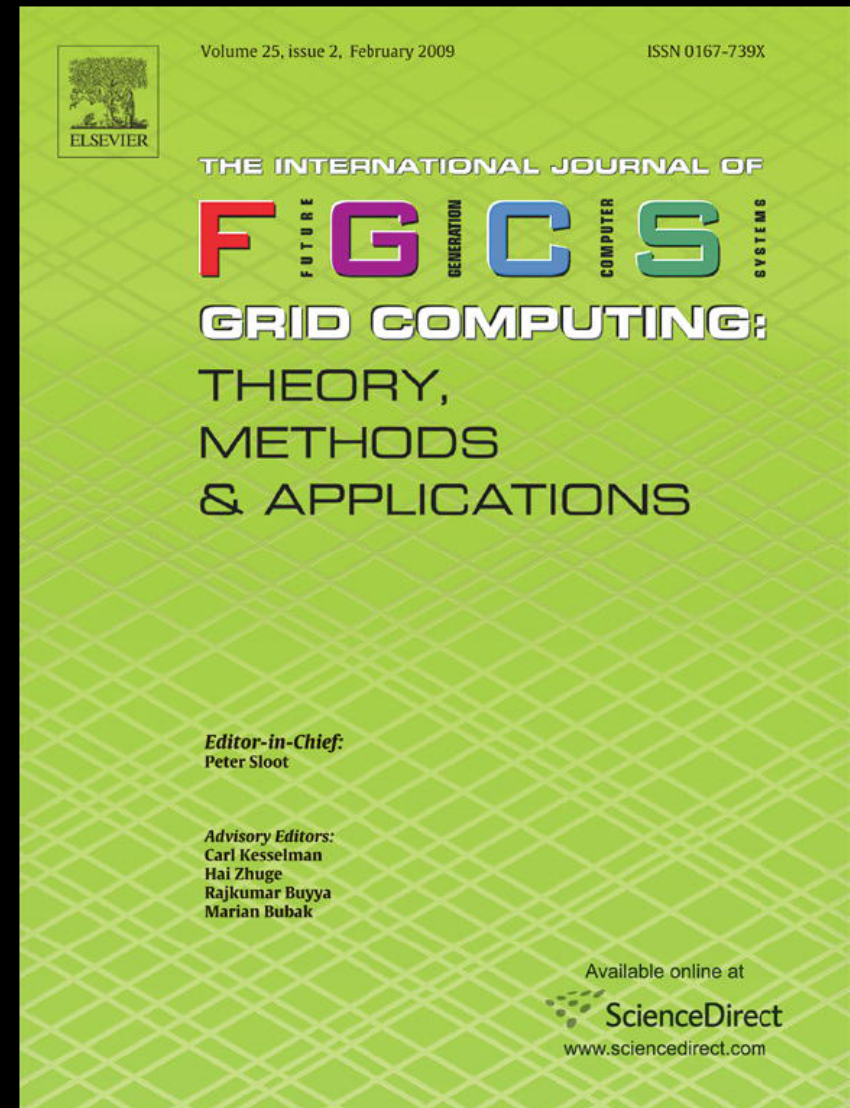
Last Thoughts

- Energy consumption is the main issue
- Cloud Computing as solution
- We did Hybrid networking
 - now hybrid computing, what else?
- Network photonics developments
- GreenSonar (aka PerfSonar)
- Smart energy conscious infrastructure

Need for Scientific Publications!

Call for papers!

- Guest Editors:
Naohisa Ohta & Paul Hearty
& Cees de Laat
- Special section on CineGrid!
- 6-8 papers in a section
- Submission via:
<http://ees.elsevier.com/fgcs/>
- CineGrid section submission site is up
- Info: delaat@uva.nl
- Submission deadline March 1st 2010



The Power of Change?

OR

The Change of Power!

sc09.delaat.net
Questions ?

p.s. On teleportation: look at prof. Eric Verlinde's work on emergent phenomena relating bit density, entropy and gravity!

<http://staff.science.uva.nl/%7Eerikv/>