CLUSTERIX

Running Applications on the CLUSTERIX Grid

Roman Wyrzykowski*, Norbert Meyer**

*Czestochowa University of Technology
**Poznań Supercomputing and Networking Center

Cracow '06 Grid Workshop

Outline

- CLUSTERIX National Grid Project
  - status, goals and architecture
  - pilot installation
- CLUSTERIX middleware
- Running applications in CLUSTERIX environment
- Meta-applications in CLUSTERIX
- Testing meta-applications
  - FEM modeling of castings solidification
  - clustering by parallel differential evolution
  - prediction of protein structures
- Final remarks

Project Status

- started on January 2004
- finished on June 2006
- 12 members – Polish supercomputing centers and MANs
- total budget – 1,2 milion Euros
- 53 % funded by the consortium members, and 47 % - by the Polish Ministry of Science and Information Society Technologies

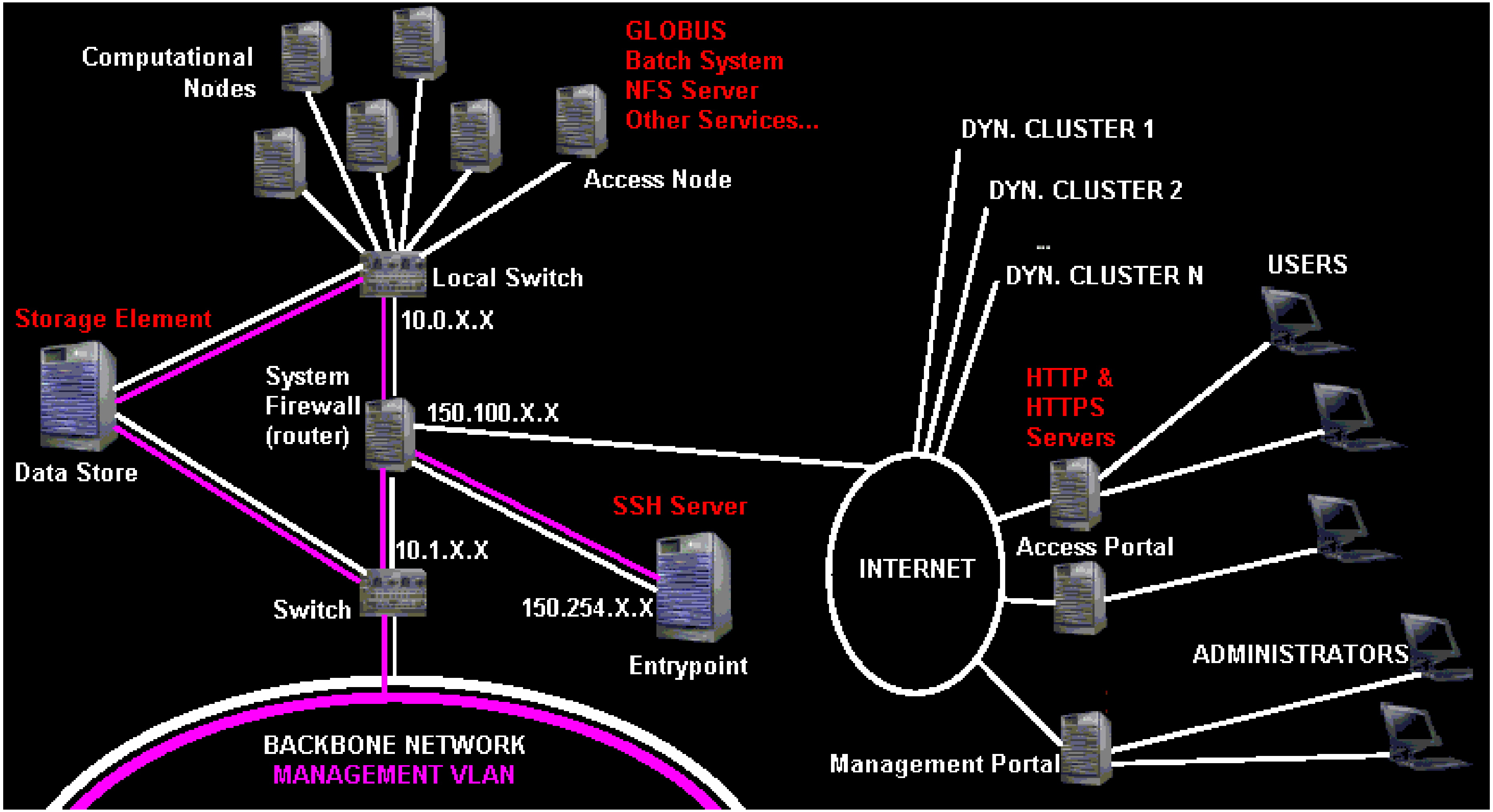
Partners

- **Częstochowa University of Technology (coordinator)**
- Poznań Supercomputing and Networking Center (PNSC)
- Academic Computing Center CYFRONET AGH, Kraków
- Academic Computing Center in Gdańsk (TASK)
- Wrocław Supercomputing and Networking Center (WCSS)
- Technical University of Białystok
- Technical University of Łódź
- Marie Curie-Skłodowska University in Lublin
- Warsaw University of Technology
- Technical University of Szczecin
- Opole University
- University of Zielona Góra

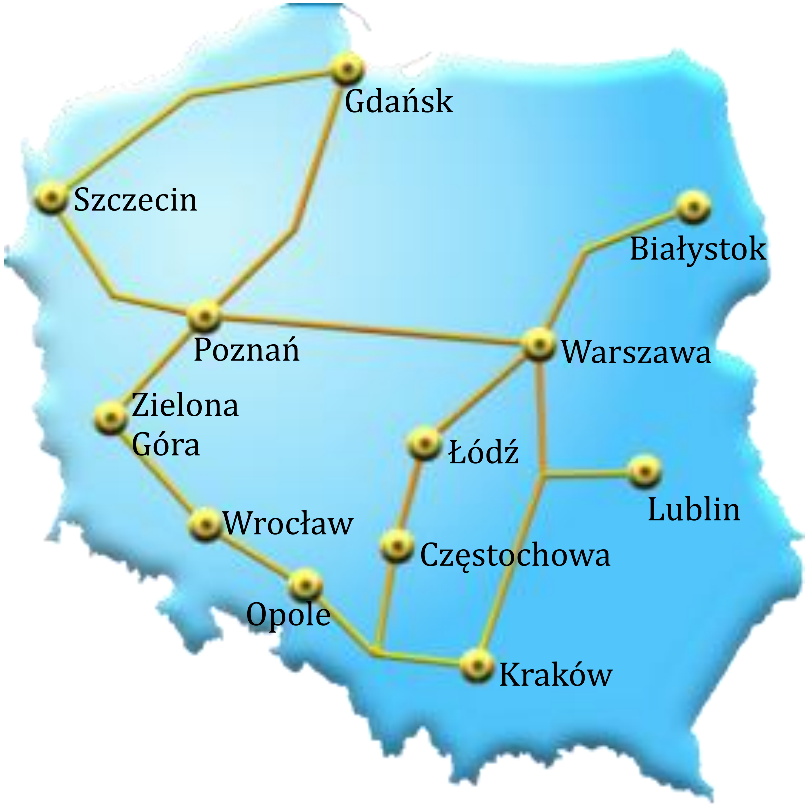
CLUSTERIX Overview

- Mechanisms and tools (middleware) that allow the deployment of a **production Grid environment**
- Basic infrastructure - local Linux PC-clusters (64-bit architecture) geographically distributed, located in independent centers connected by the fast backbone provided by the Polish Optical Network PIONIER (10 Gbps)
- Existing PC-clusters as well as anew built clusters can be dynamically connected to the basic infrastructure

CLUSTERIX Architecture



Pilot Installation



- 12 local clusters with 200+ IA-64 in the core
- Linux Debian, kernel 2.6.x
- PIONIER Network: 3000+ km of fibers with 10Gbps DWDM technology
- 2 VLANs with dedicated 1Gbps bandwidth for the CLUSTERIX network
- whole network has dual-stack network with IPv4 and IPv6 fully enabled

Middleware in CLUSTERIX

- CLUSTERIX middleware is based on Globus Toolkit 2.4 plus web services - with Globus 2.4 available in Globus 3.2 distribution

  - this makes the created software easier to reuse

  - allows for interoperability with other Grid systems on the service level

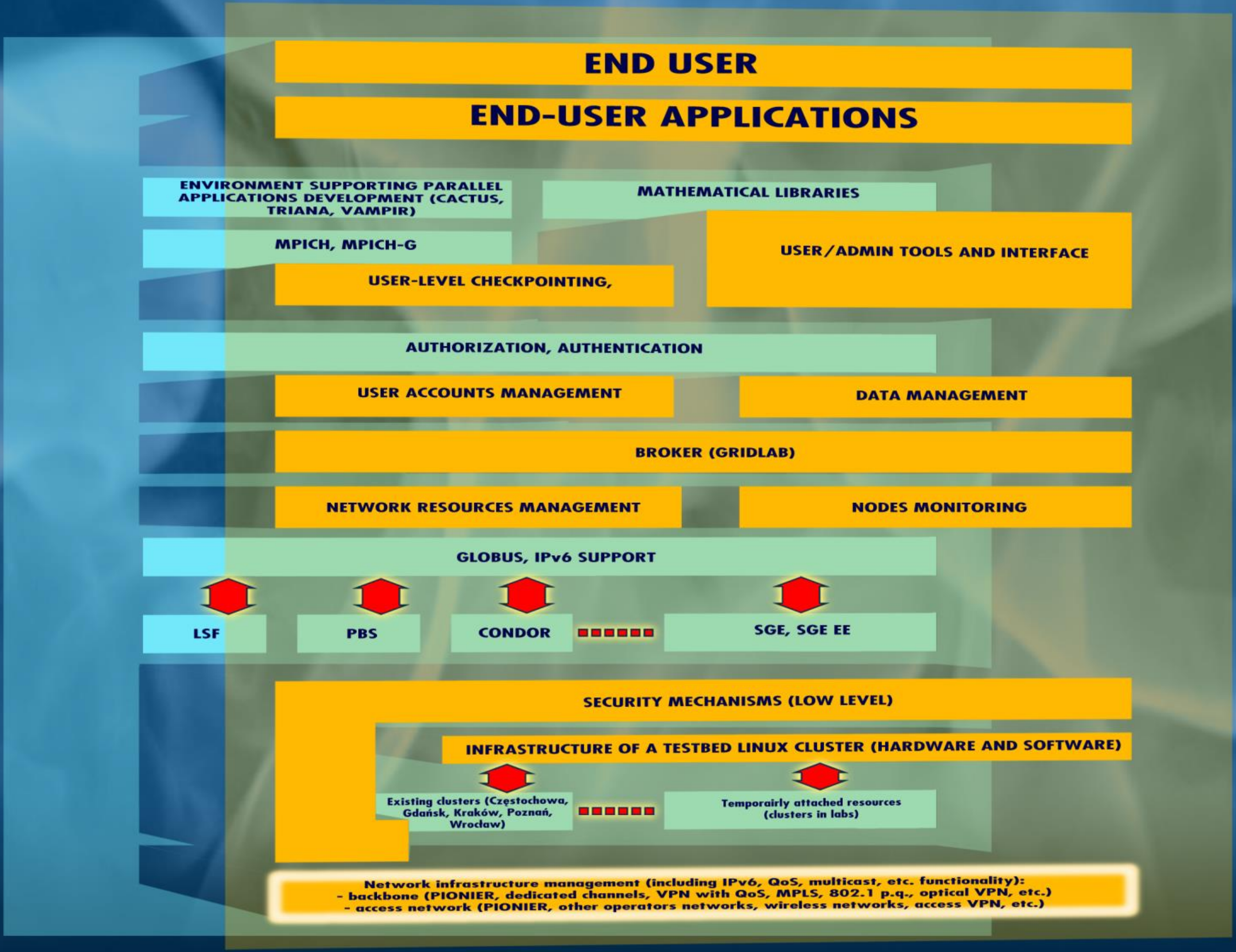
- *Open Source* technology, including LINUX (Debian, kernel 2.6.x) and batch systems (Open PBS/Torque)

  - open software is easier to integrate with existing and new products
  - allows anybody to access the project source code, modify it, and publish the changes

  - makes the software more reliable and secure

- Existing middleware is used extensively in the CLUSTERIX project, e.g., GRMS from *GridLab*

SOFTWARE ARCHITECTURE



END USER

END-USER APPLICATIONS

ENVIRONMENT SUPPORTING PARALLEL APPLICATIONS DEVELOPMENT (CACTUS, TRIANA, VAMPIR)

MATHEMATICAL LIBRARIES

MPICH, MPICH-G

USER-LEVEL CHECKPOINTING,

USER/ADMIN TOOLS AND INTERFACE

AUTHORIZATION, AUTHENTICATION

USER ACCOUNTS MANAGEMENT

DATA MANAGEMENT

BROKER (GRIDLAB)

NETWORK RESOURCES MANAGEMENT

NODES MONITORING

GLOBUS, IPv6 SUPPORT

LSF

PBS

CONDOR

SGE, SGE EE

SECURITY MECHANISMS (LOW LEVEL)

INFRASTRUCTURE OF A TESTBED LINUX CLUSTER (HARDWARE AND SOFTWARE)

Existing clusters (Częstochowa, Gdańsk, Kraków, Poznań, Wrocław)

Temporairly attached resources (clusters in labs)

Network infrastructure management (including IPv6, QoS, multicast, etc. functionality):
- backbone (PIONIER, dedicated channels, VPN with QoS, MPLS, 802.1 p.q., optical VPN, etc.)
- access network (PIONIER, other operators networks, wireless networks, access VPN, etc.)

GRMS: Resource Management System

- GRMS is an open source scheduling system for large scale distributed computing infrastructures
- Designed to deal with resource management challenges in Grid environments:
  - setting up execution environments before and after job execution
  - remote job submission and controlling
  - files staging
  - load-balancing among clusters
  - more
- Based on the dynamic resource selection, mapping and advanced grid scheduling methodologies, combined with feedback control architecture

GRMS features developed in CLUSTERIX

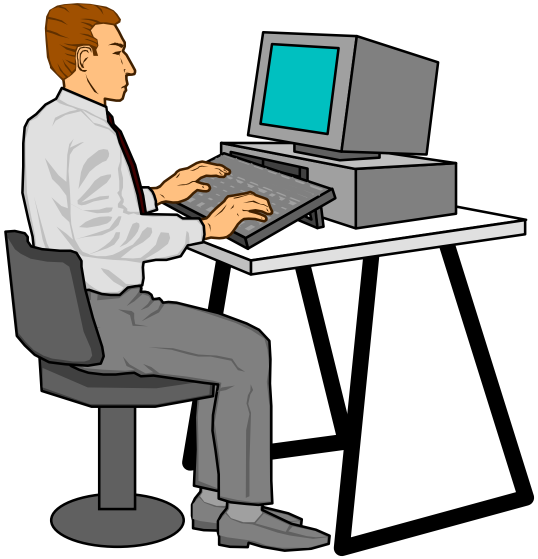
- Support for distributed MPICH-G2 application
  - allows users to submit jobs which will be dispersed among many nodes of many clusters,
  - makes CLUSTERIX able to execute large, multi-process applications
- Prediction of Job execution
  - Increases the resource management efficiency by providing estimated values
  - Allows resource broker to find out:
    - job execution time
    - job pending time in given queue
    - probable resource utilization by the job
    - estimation of inaccuracy

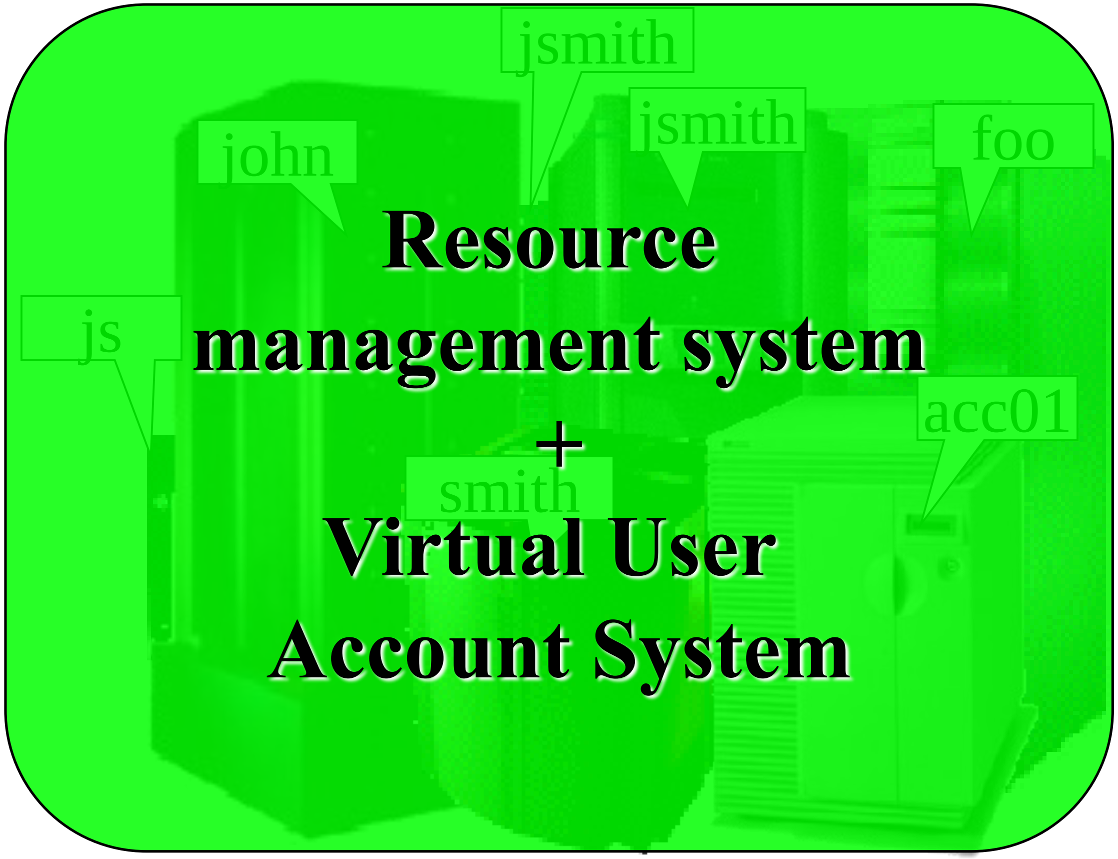
CDMS: CLUSTERIX Data Management System

- Goals of design:
  - transparent access: convenient API for client applications
  - reliability: data replication, distributed Data Broker
  - security and safety of transferred and stored data: user authentication/authorization (GSI based), data encryption permissions delegation, Access Control Lists embedded in metadata
  - ability to transparently compress data
  - access optimization: Statistic and Optimization Subsystems
- Basic technologies: gridFTP and GSI from Globus 2.4, web services implemented using gSOAP and GSI plugin from GridLab

Virtual Users' Accounts System - VUS

Normally the user has had to apply for account on each machine





Resource management system

+

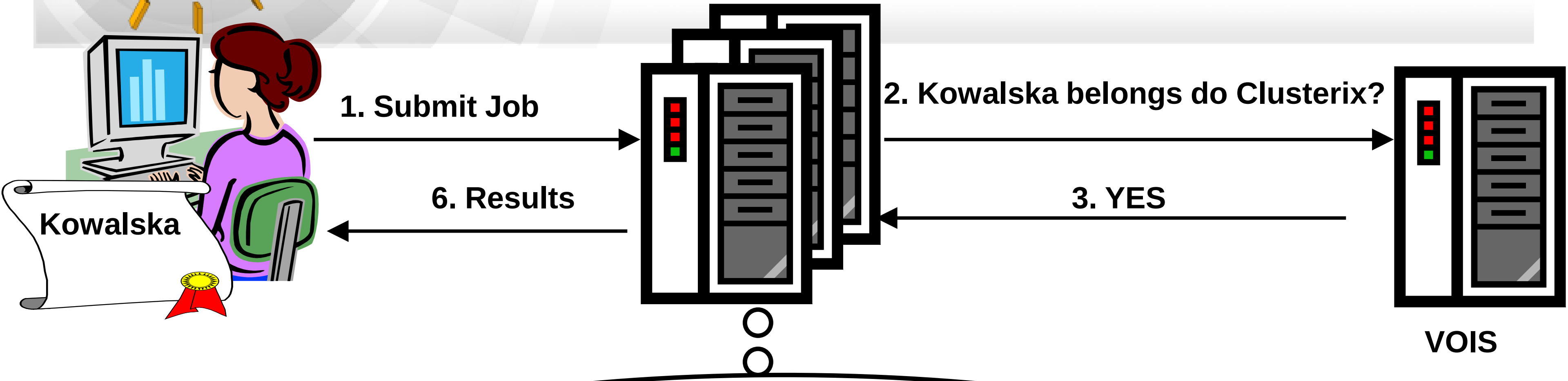
Virtual User Account System

Virtual Users' Accounts System (cont.)

- Set of generic accounts that can be assigned to consecutive jobs
- The user is authenticated, authorized and then logged on a 'virtual' account (one user per one account at the time)
- Allows running user's jobs without having an user account on a node (or local cluster)

- Decreases management (administration overheads)
- Full accounting information
- Keeps local and global policies
- Supports different *grid players*: user, resource owner, organization manager

VUS in Action

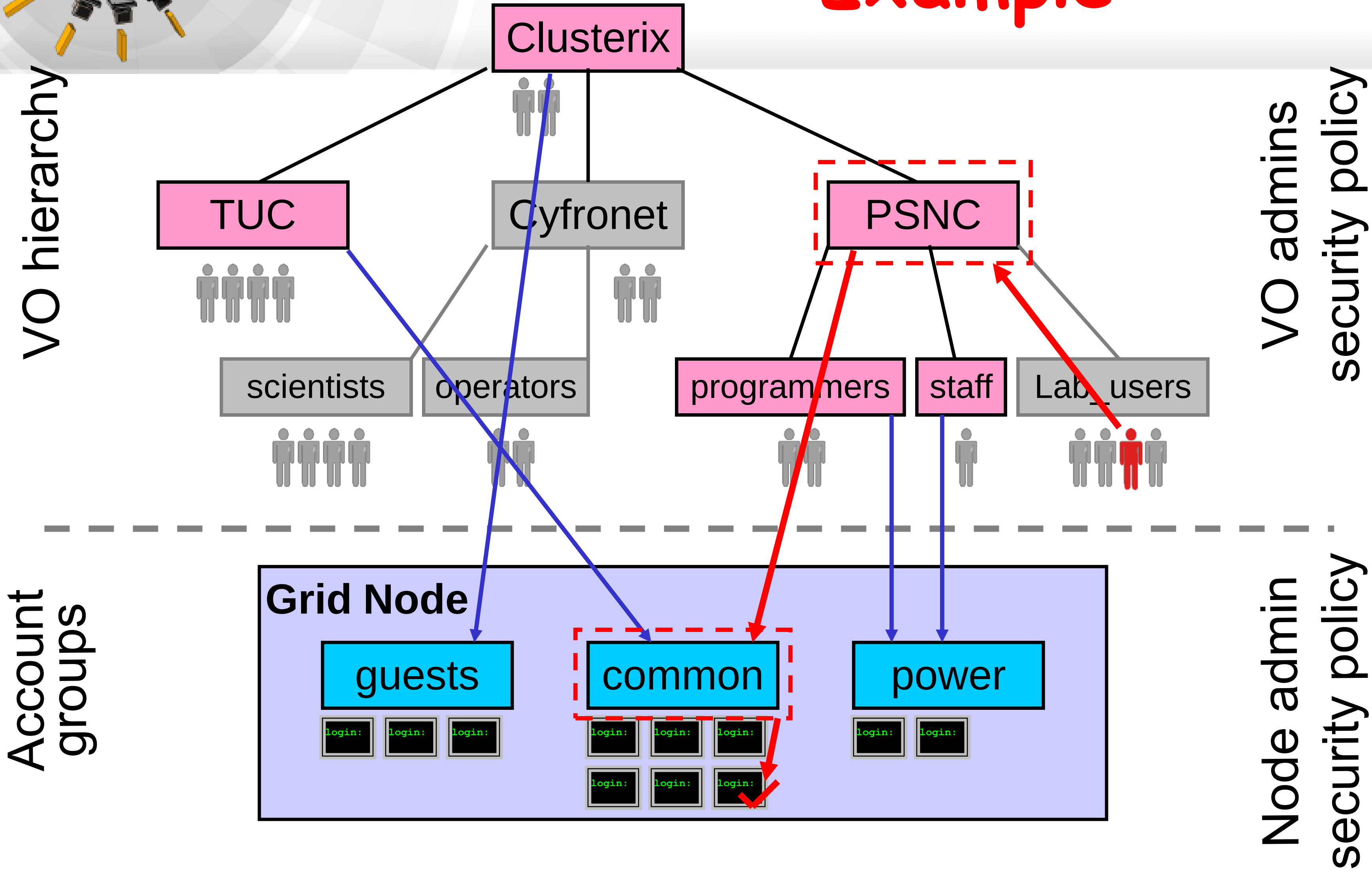


4. User is logged on a CLUSTERIX virtual account (1 user per 1 account at the moment)

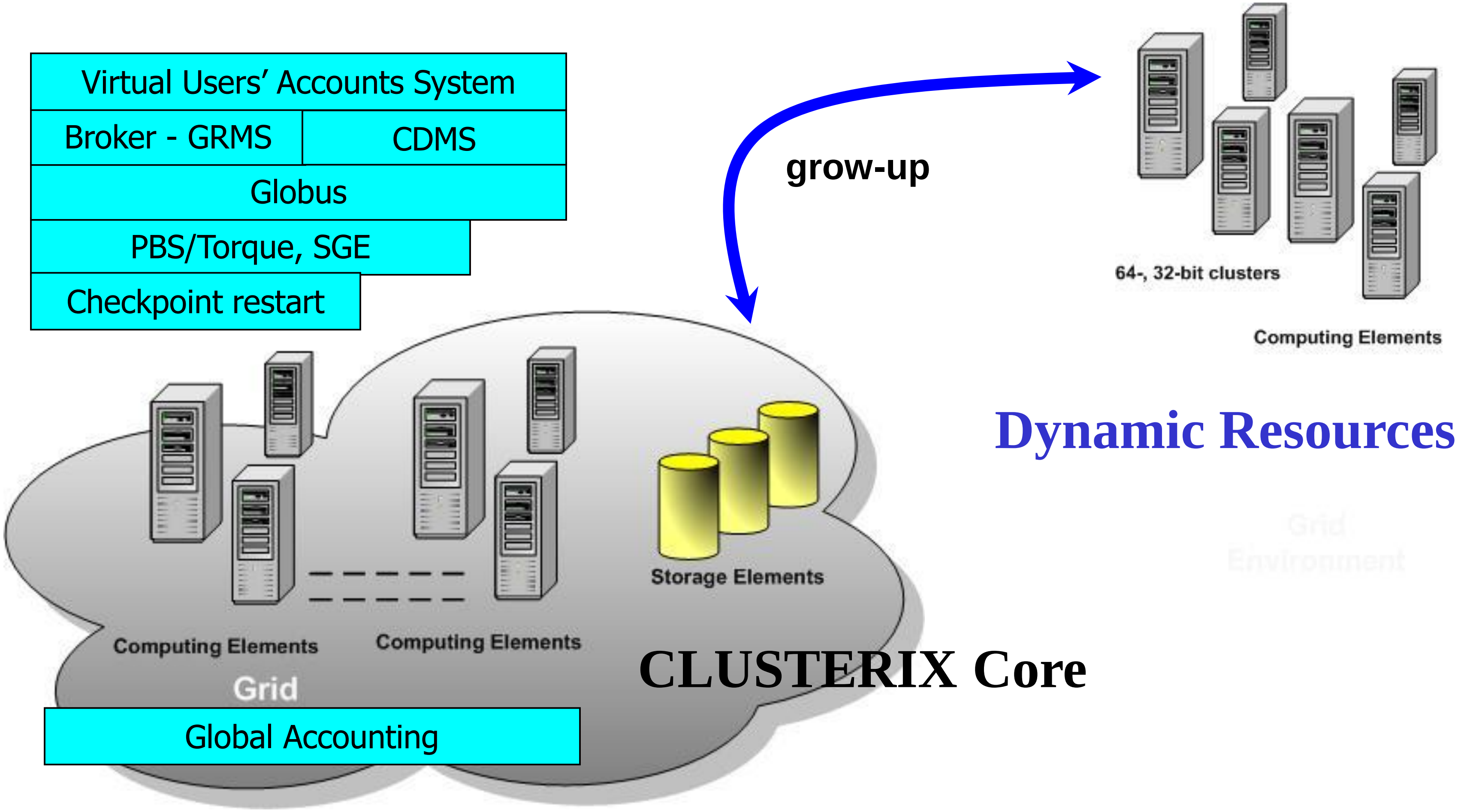
5. Execute Job

7. If an account is not used, another user could be logged on this account

VOIS Authorization - Example



Integrating Dynamic Clusters



Virtual Users' Accounts System

Broker - GRMS | CDMS

Globus

PBS/Torque, SGE

Checkpoint restart

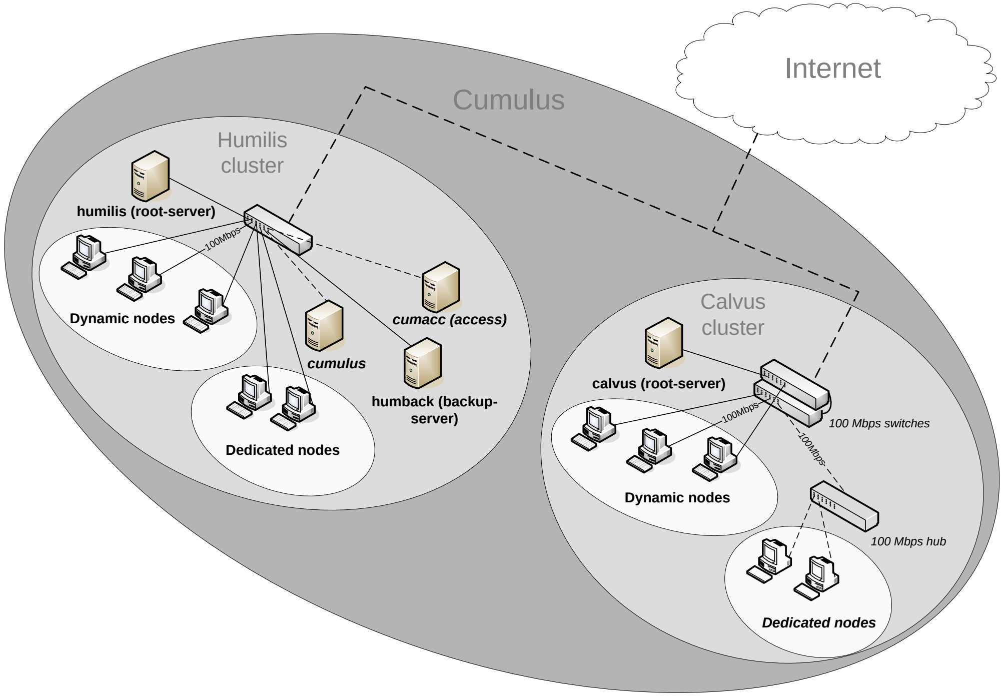
Global Accounting

Dynamic Resources

CLUSTERIX Core

Integrating Dynamic Clusters (cont.)

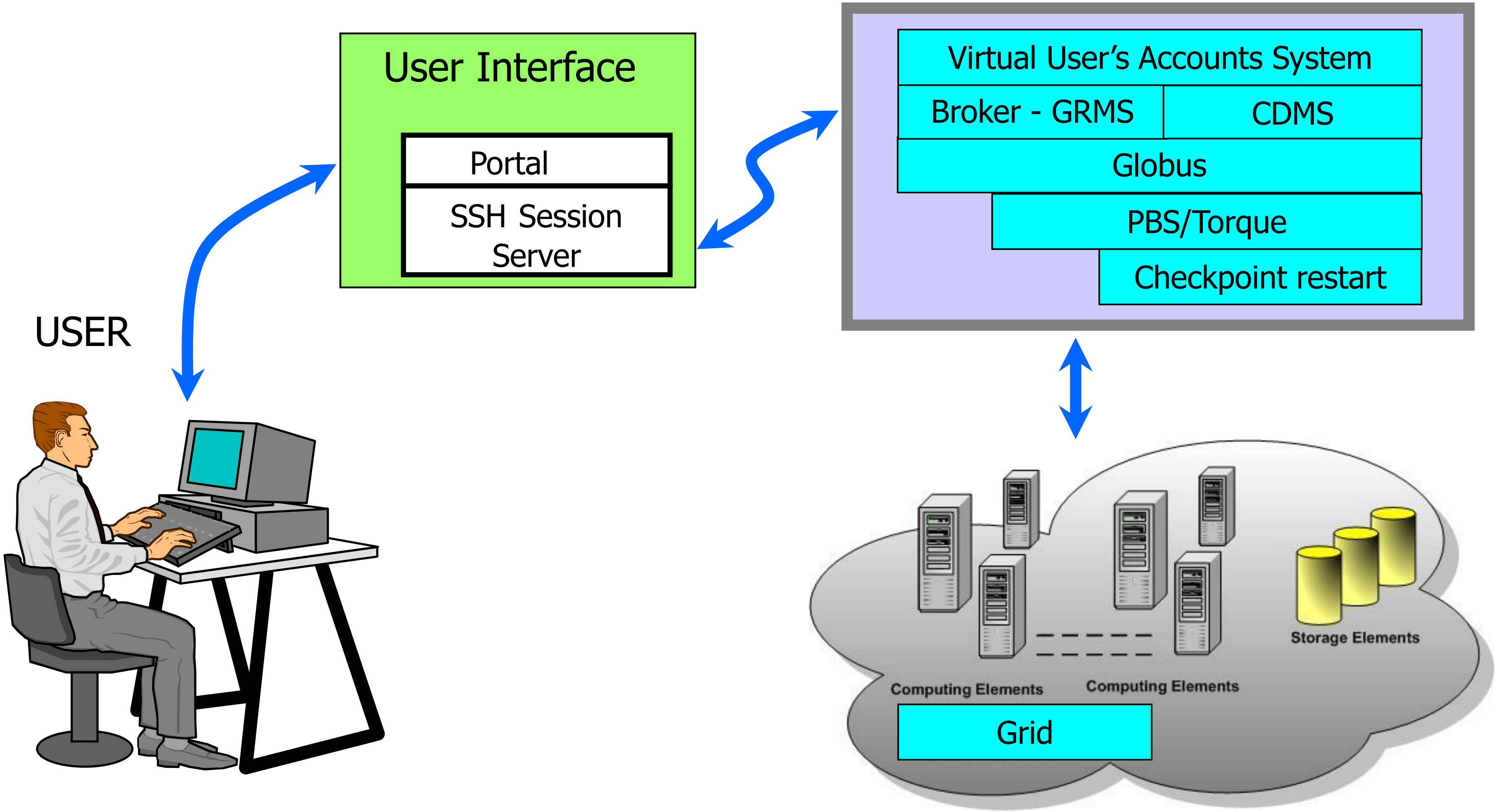
- Ability to connect dynamic clusters from anywhere (clusters from campuses and universities)
- Utilize external clusters during nights or non-active periods
- Make CLUSTERIX infrastructure scalable



Pilot Applications

- Selected scientific applications (out of ~30) have been developed for experimental verification of the project assumptions and results, as well as to achieve real application results
- Running both HTC applications, as well as large-scale distributed HPC applications that require parallel use of one or more local clusters (meta-applications)
- Two directions:
  - adaptation of existing applications for Grids
  - development of new applications

Task execution in CLUSTERIX



User Interface

Portal

SSH Session Server

USER

Virtual User's Accounts System

Broker - GRMS

CDMS

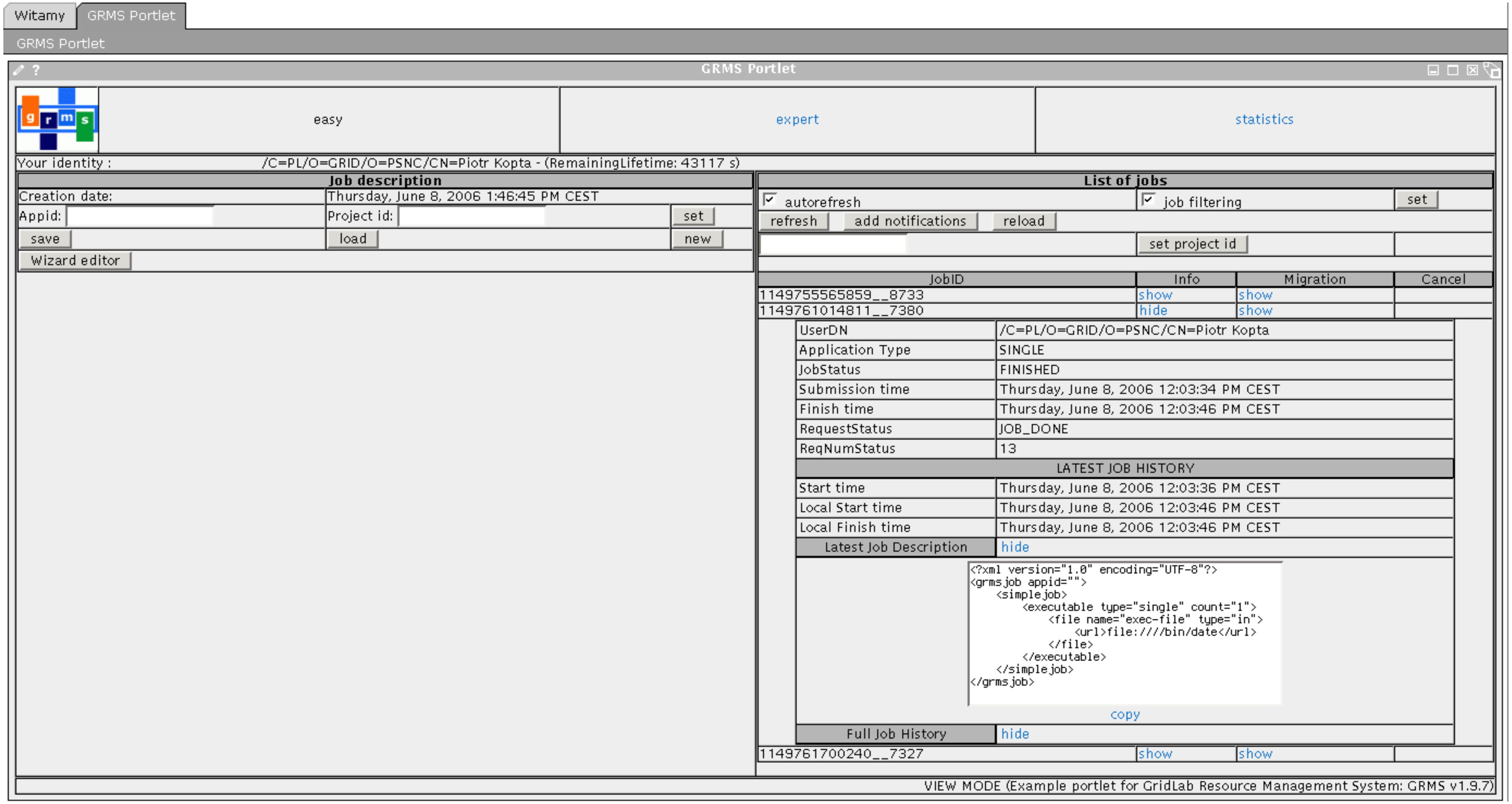
Globus

PBS/Torque

Checkpoint restart

Grid

GRMS Portal



Your identity : /C=PL/O=GRID/O=PSNC/CN=Piotr Kopta - (RemainingLifetime: 43117 s)

Job description

Creation date:	Thursday, June 8, 2006 1:46:45 PM CEST	
Appid:	Project id:	set
save	load	new
Wizard editor		

List of jobs

☑ autorefresh ☑ job filtering set

refresh add notifications reload

set project id

JobID	Info	Migration	Cancel
1149755565859__8733	show	show	
1149761014811__7380	hide	show	

UserDN	/C=PL/O=GRID/O=PSNC/CN=Piotr Kopta
Application Type	SINGLE
JobStatus	FINISHED
Submission time	Thursday, June 8, 2006 12:03:34 PM CEST
Finish time	Thursday, June 8, 2006 12:03:46 PM CEST
RequestStatus	JOB_DONE
ReqNumStatus	13
LATEST JOB HISTORY	
Start time	Thursday, June 8, 2006 12:03:36 PM CEST
Local Start time	Thursday, June 8, 2006 12:03:46 PM CEST
Local Finish time	Thursday, June 8, 2006 12:03:46 PM CEST
Latest Job Description	hide

```
<?xml version="1.0" encoding="UTF-8"?>
<grmsjob appid="">
    <simplejob>
        <executable type="single" count="1">
            <file name="exec-file" type="in">
                <url>file:///bin/date</url>
            </file>
        </executable>
    </simplejob>
</grmsjob>
```

copy

Full Job History	hide

JobID	Info	Migration	Cancel
1149761700240__7327	show	show	

VIEW MODE (Example portlet for GridLab Resource Management System: GRMS v1.9.7)

```xml
<grmsjob appid="psolidify">
    <simplejob>
        <resource>
            <localrmname>pbs</localrmname>
        </resource>
        <executable type="mpi" count="8">
            <file name="exec" type="in">
                <url>gsiftp:////access.wcss.clusterix.pl/~/myapp/psolidify/</url>
            </file>
            <arguments>
                <value>250000.prl</value>
                <file name= "250000.prl" type="in">
                    <url>gsiftp://access.wcss.clusterix.pl/~/data/250000.prl</url>
                </file>
                </arguments>
                <stdout>
                    <url>gsiftp://access.wcss.clusterix.pl/~/app1.out</url>
                </stdout>
        </executable>
    </simplejob>
</grmsjob>
```

Basic scenario of Job execution in CLUSTERIX:

- The user submits the Job to GRMS through the portal, providing Job Description
- GRMS chooses the best resource for the Job execution, according to Job Description (hardware and software)
- Staging:
  - a) executables (also scripts)
  - b) input data described by logical or physical URL, from CDMS - CLUSTERIX Data Management System
- VUS is responsible for mapping the user credentials onto physical accounts in the local clusters
- Job execution
- After finishing the Job, output results are picked up and transferred to CDMS; then physical accounts are cleaned out by VUS

Different scenarios of using Grid resources

- Grid as the resource pool

  an appropriate computational resource (local cluster) is found via resource management system, and the sequential application is started there

- Parallel execution on grid resources (meta-applications):
  - single parallel application being run on geographically distributed resources
  - Grid-aware parallel application - the problem is decomposed taking into account Grid architecture

MPICH-G2

- The MPICH-G2 tool is used as a grid-enabled implementation of the MPI standard
- It is based on the Globus Toolkit used for such purposes as authentication, authorization, process creation, process control, …
- MPICH-G2 allows to couple multiple machines, potentially of different architectures, to run MPI applications
- To improve performance, it is possible to use other MPICH-based vendor implementations of MPI in local clusters (e.g. MPICH-GM)

```
<grmsjob appid="mpichg2test" persistent="true">
    <simplejob>
        <resource>
            <hostname tileSize=„8">access.pcss.clusterix.pl</hostname>
            <localrmname>pbs</localrmname>
        </resource>
        <resource>
            <hostname tileSize=„8">access.pcz.clusterix.pl</hostname>
            <localrmname>pbs</localrmname>
        </resource>
        <executable type="mpichg" count=„16">
            <file name="clx" type="in">
                <url>file:////tmp/clx_ia64_g2</url>
            </file>
            <arguments>
                <value>HOME/CLX/var/grms_demo2</value>
                <value>25</value>
                <value>1</value>
            </arguments>
            <stdout>
                <url>gsiftp://access.pcss.clusterix.pl/~/demo2.out</url>
            </stdout>
        </executable>
    </simplejob>
</grmsjob>
```

CLUSTERIX as a heterogeneous system

- Hierarchical architecture of CLUSTERIX

	latency	bandwidth	# processors
single node (MPICH-G2)		5,4 Gb/s	2
local cluster (vendor MPI)	104 µs	752 Mb/s	6-32
local cluster (MPICH-G2)	124 µs	745 Mb/s	6-32
meta-cluster (MPICH-G2)	10 µs	33 Mb/s	up to 200

- It is not a trivial issue to adapt an application for its efficient execution in the meta-cluster environment
- Communicator construction in MPICH-G2 can be used to represent hierarchical structures of heterogeneous systems, allowing applications to adapt their behavior to such architectures

NuscaS

Czestochowa University of Technology

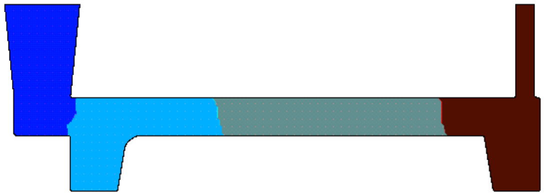
Tomasz Olas

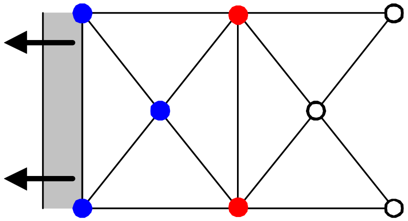
Application areas:

- **FEM simulation of different thermo-mechanic phenomena:**

  heat transfer, kinetics of solidification in castings, stresses in thermo-elasto-plastic states, hot-tearing in castings, mechanical interactions between bodies, damage, etc.

NuscaS package: Parallelization





● węzły wewnętrzne

● węzły brzegowe

○ węzły zewnętrzne

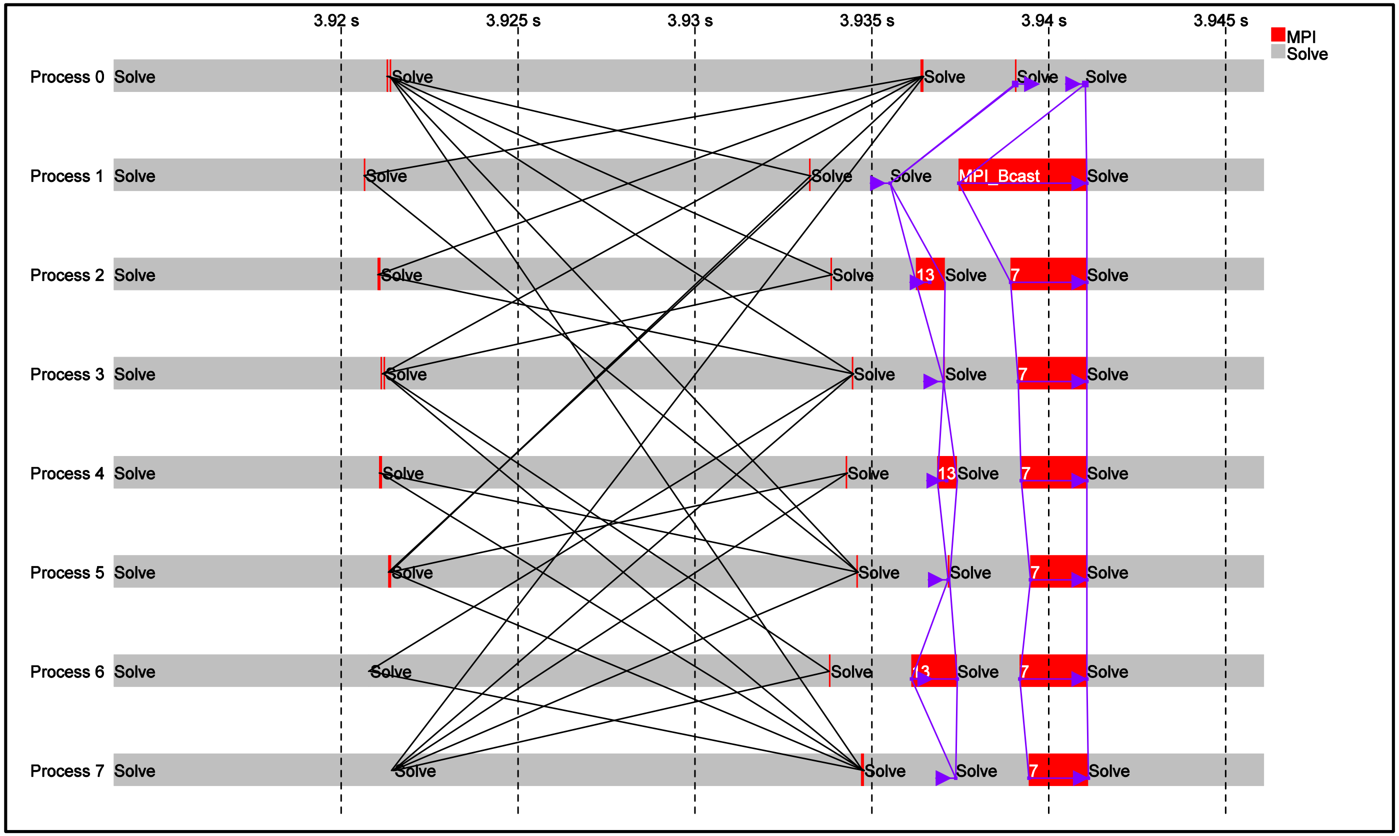
budowanie
lokalnego układu równań

$$
\underset{\begin{array}{ccc}\text{węzły wewnętrzne} & \text{węzły brzegowe} & \text{węzły zewnętrzne}\end{array}}{\overset{A_i}{\left[\begin{array}{c:c|c} A^{ii} & A^{ib} & 0 \\ \hdashline A^{bi} & A^{bb} & A^{be} \end{array}\right]}}
\times
\begin{array}{cc} & x_i \\ \text{węzły wewnętrzne} & x^i \\ \text{węzły brzegowe} & x^b \\ \text{węzły zewnętrzne} & x^e \end{array}
=
\begin{array}{cc} & b_i \\ \text{węzły wewnętrzne} & b^i \\ \text{węzły brzegowe} & b^i \end{array}
$$

Solving linear systems in parallel (1)

- Conjugate Gradient (CG) method is used
- A version of the CG algorithm (proposed by Meisel & Meyer) with only one point of synchronization is exploited to reduce idle time of processors
- Matrix-vector multiplication with sparse matrices is chosen as a computational kernel
- Overlapping of computation and communication facilitates hiding communication latencies

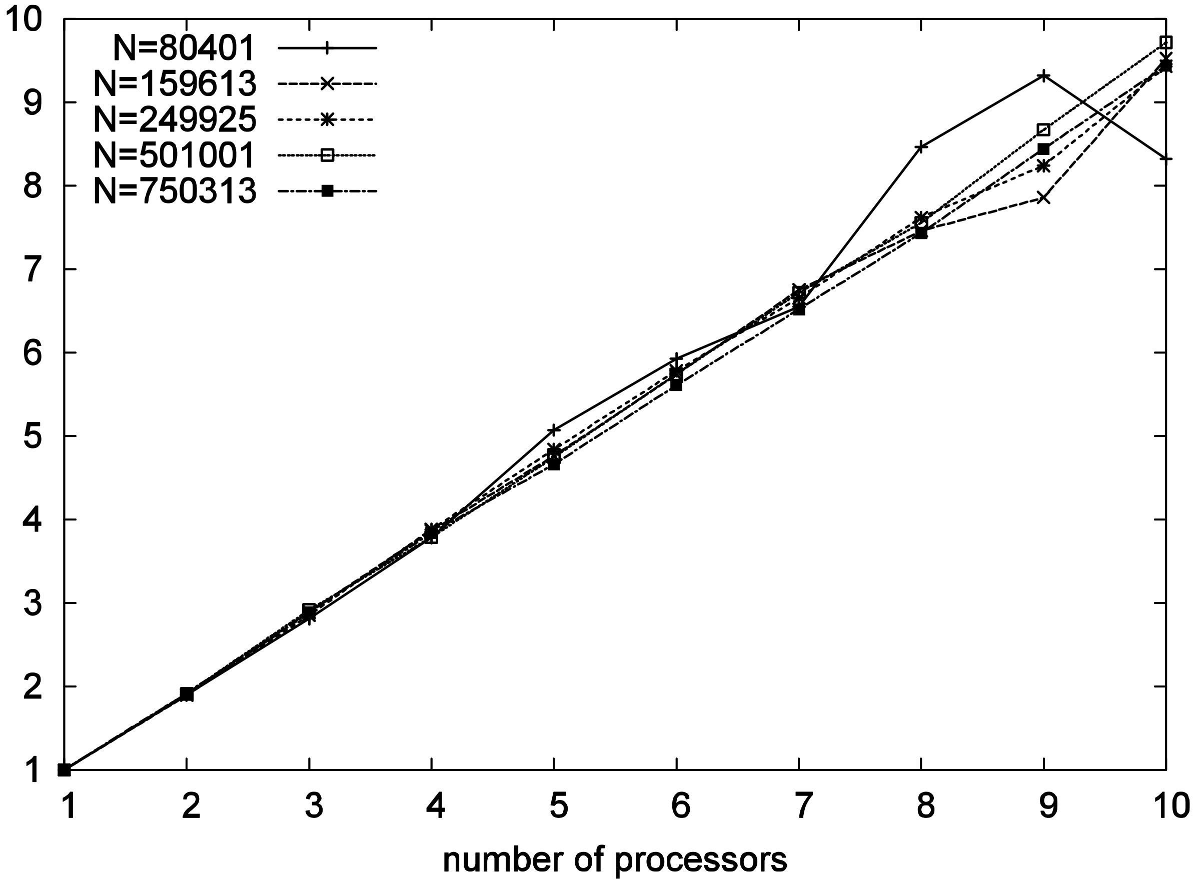
Solving linear systems in parallel (2)



250000.8.8.bpv: Global Timeline (3.914 s - 3.946 s = 32.489 ms)

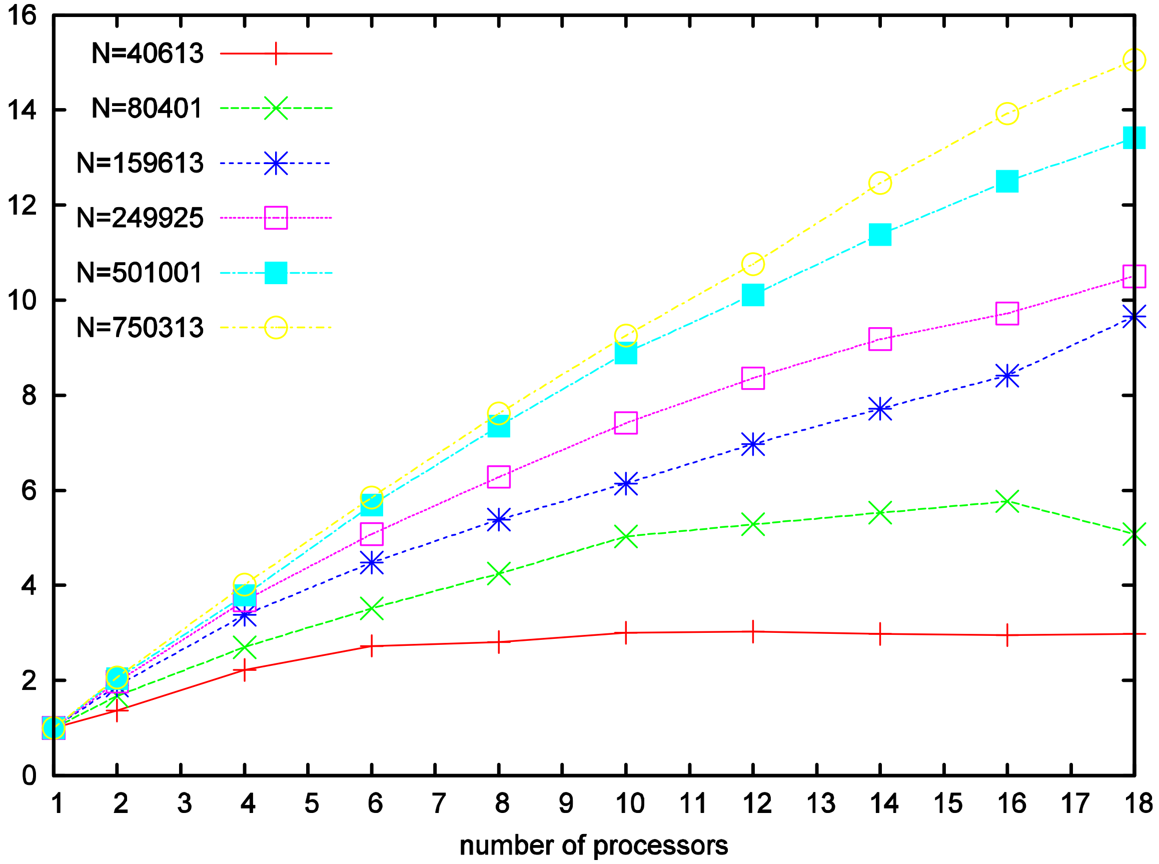
NuscaS

Single-site Performance



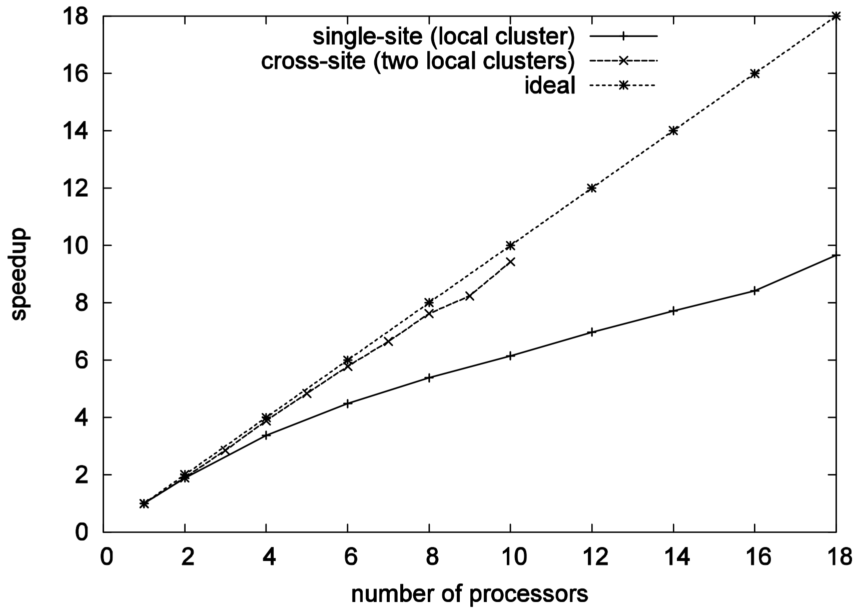
NuscaS

Cross-site Performance

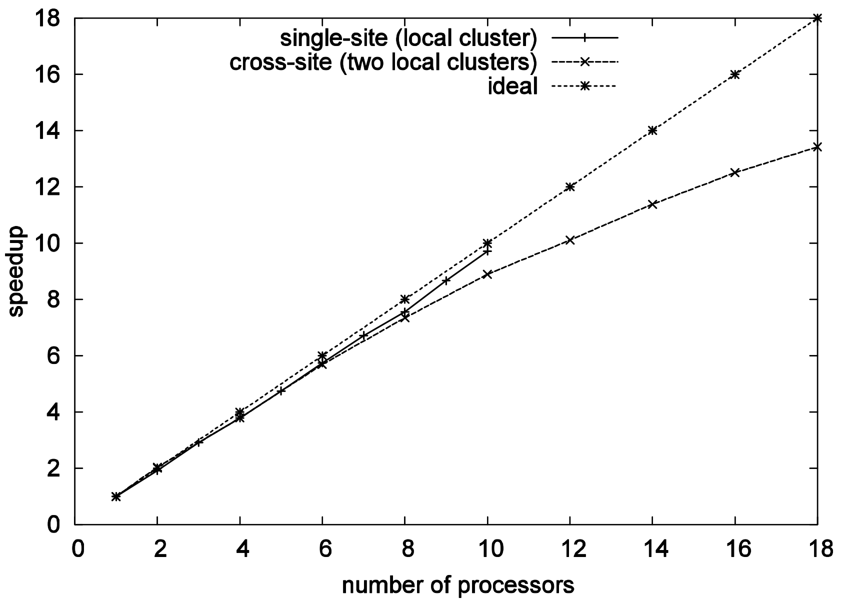


Cross-site versus Single-site Performance

249 925 nodes



501 001 nodes



ClustPDE:

Clustering by Parallel Differential Evolution

Białystok Technical University

Wojciech Kwedlo

Application areas:

- data mining, market analysis, vector quantization

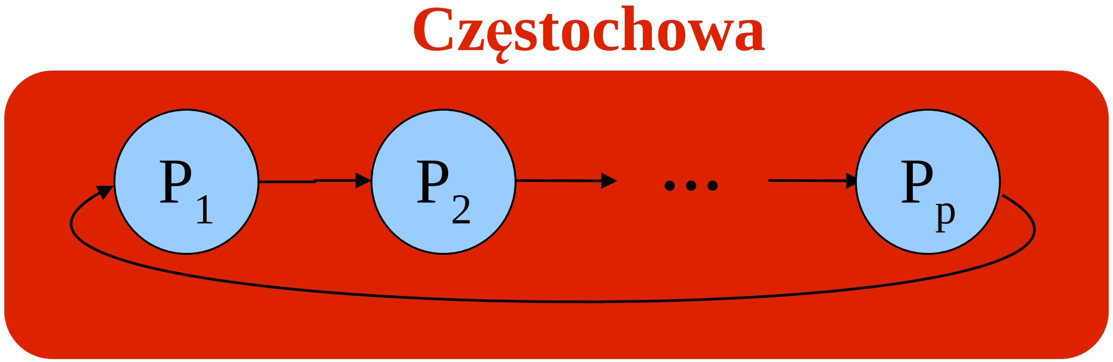
ClustPDE: Introduction

- The goal of clustering is to divide the learning set of M feature vectors from R^N into k groups, in order to minimize intra-group and maximize inter-group differences.

- Since standard algorithms (k-means) are easily being trapped in local optima, we use *differential evolution* (a global optimization method) to solve this problem.

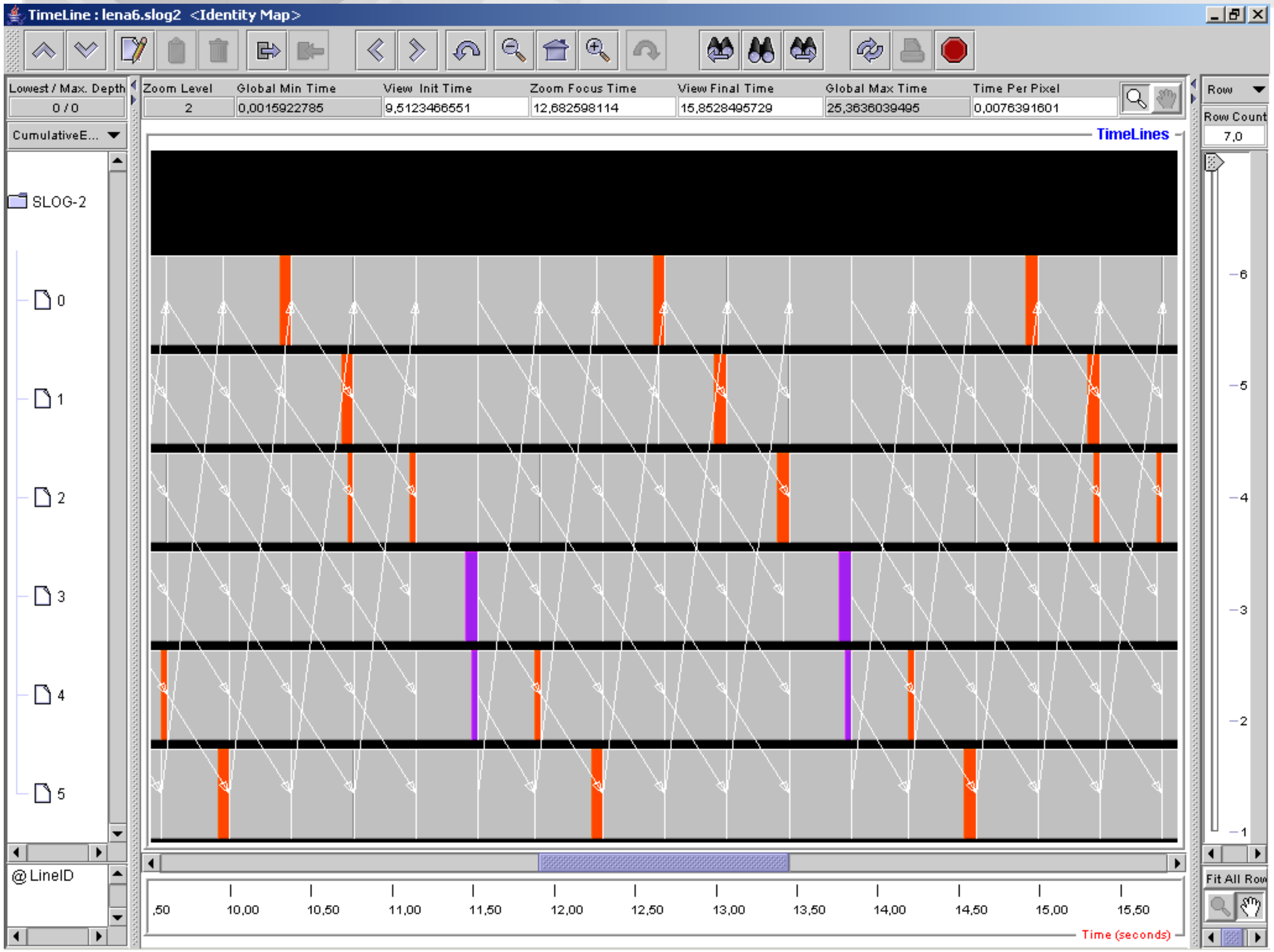
- However evolutionary algorithms demand a lot of computing power, hence parallelization is necessary.

ClustPDE:
single cluster setup



- In this application, processes form a ring-based pipeline
- The use of asynchronous (MPI_ISend/IRecv) communication allows us to hide communication costs
- All processes in a single cluster (Częstochowa)

ClustPDE – application trace for 6 CPUs



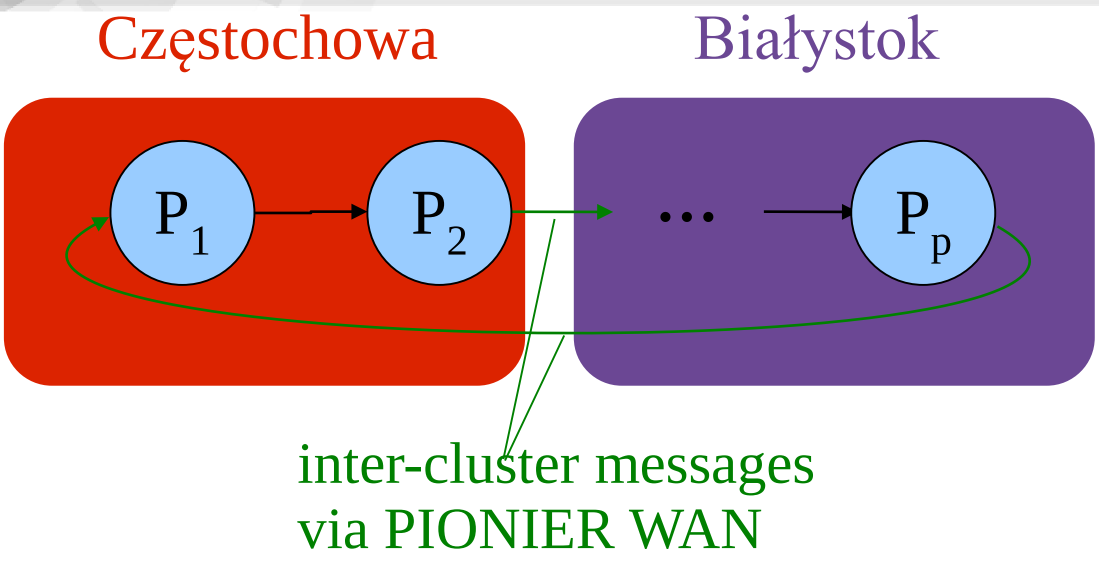
TimeLine : lena6.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel	Row Count
0 / 0	2	0,0015922785	9,5123466551	12,682598114	15,8528495729	25,3636039495	0,0076391601	7,0

Latency hiding

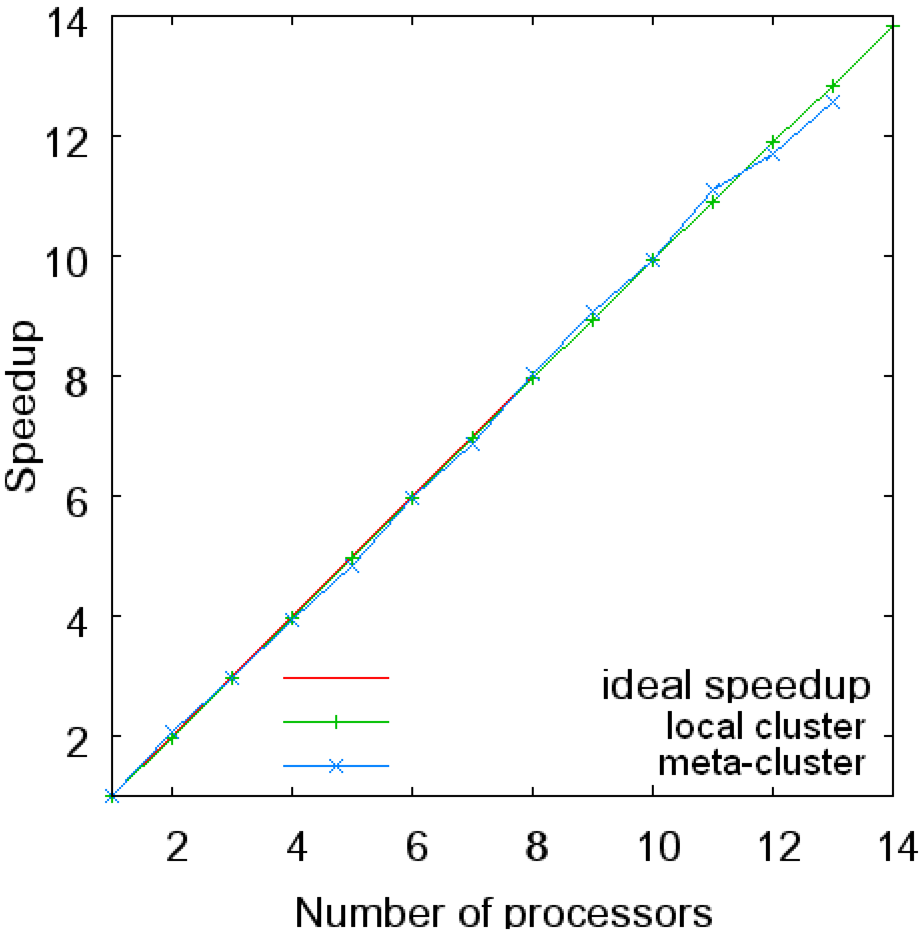
(computation simultaneous with communication)

ClustPDE:
Meta-cluster setup



- Processes divided 50%-50% between Częstochowa and Białystok
- Communication via WAN minimized (only 2 messages out of total N messages per iteration)

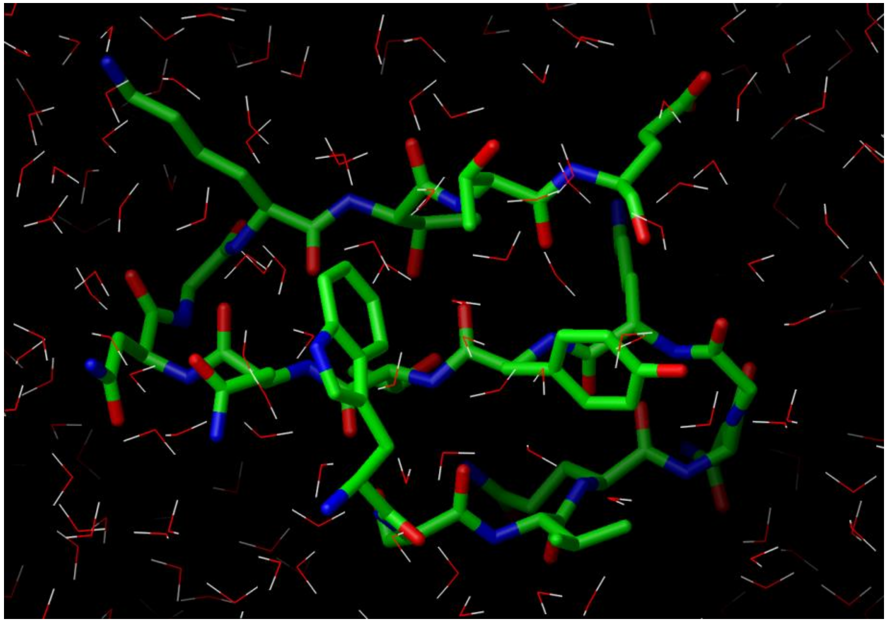
ClustPDE – speedup comparison

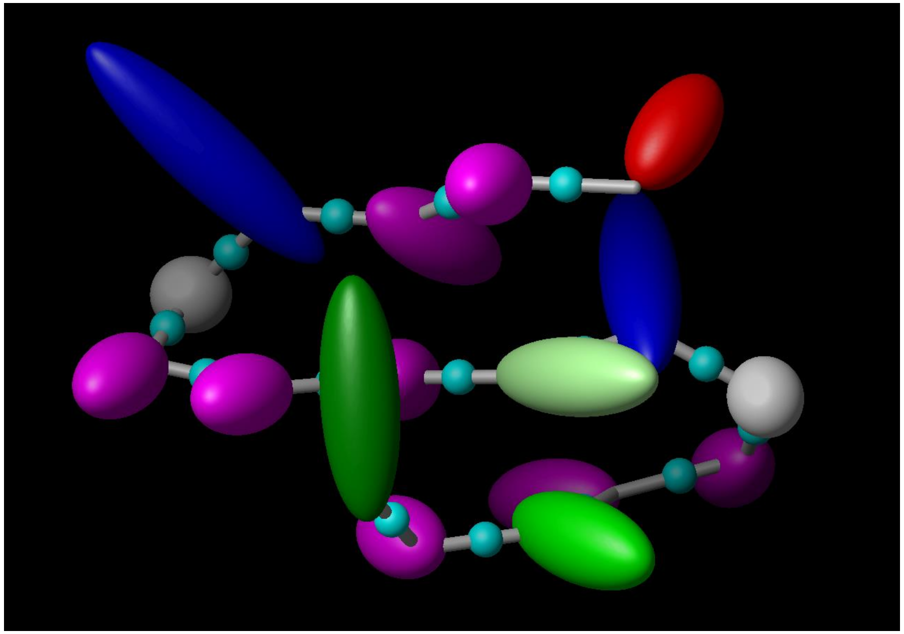


- Speedup almost linear
- Results obtained on the meta-cluster almost the same as those obtained on a single local cluster

Prediction of Protein Structures

Cezary Czaplewski, Stanisław Ołdziej, Adam Liwo
Department of Chemistry, University of Gdansk

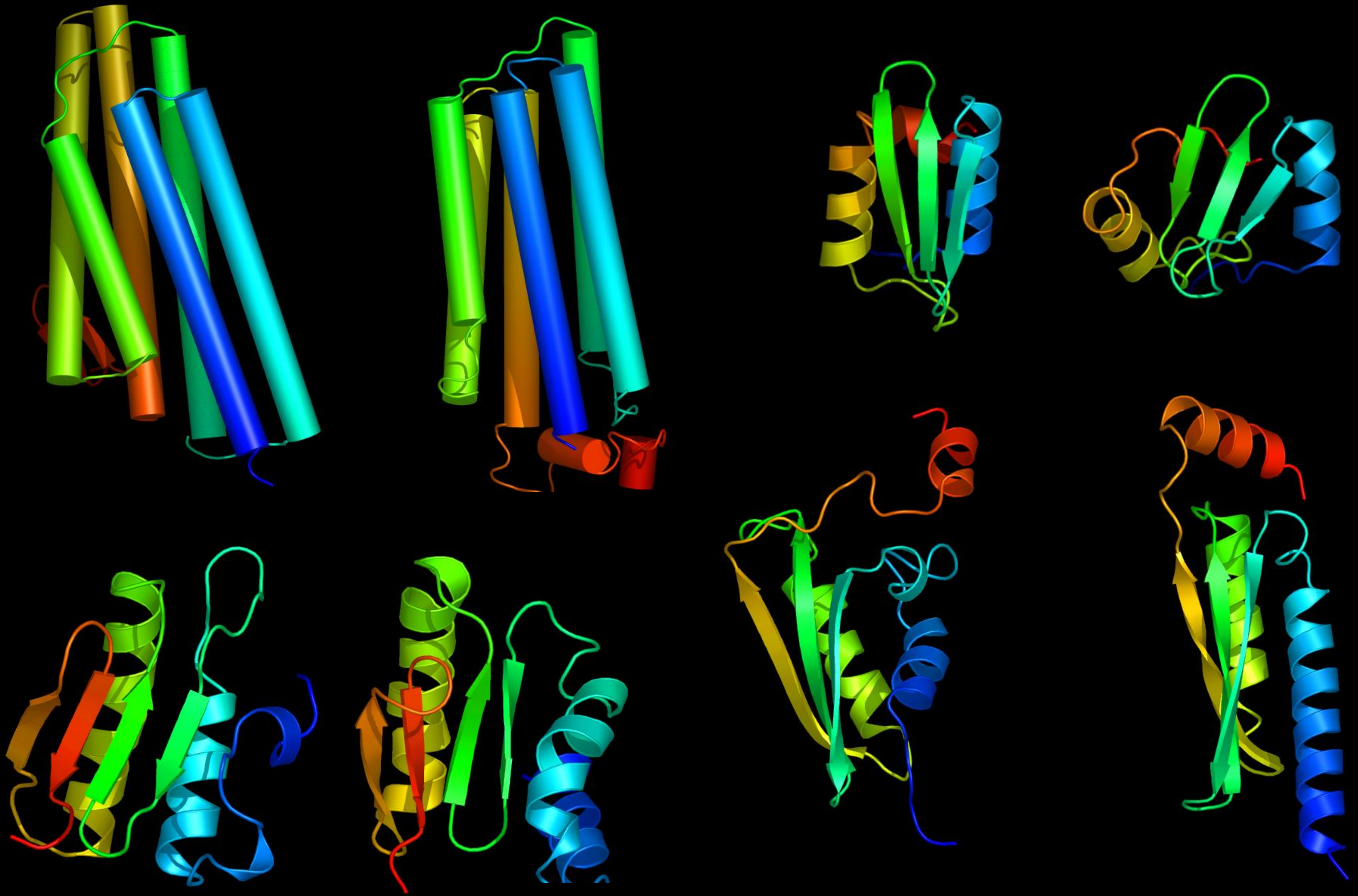




Selected UNRES/CSA results from 6th Community Wide Experiment on the

Critical Assessment of Techniques for Protein Structure Prediction

December 4-8, 2004



left - experimental structure, right - predicted structure

Prediction of Protein Structure

Performance results (1)

TASK		PB + PCz		TASK+PB+PCz	
p	time [s]	p	time [s]	p	time [s]
2	5394	1+1	5483		
4	1752	2+2	1837	2+1+1	2083
8	767	4+4	777	4+2+2	1013
12	476	6+6	500	6+3+3	616
16	351			8+4+4	456
32	174			20+6+6	199

Prediction of Protein Structure

Performance results (2)

TASK + PB + PCZ + WCSS	
p	time [s]
0 + 6 + 6 + 0	495
6 + 6 + 0 + 0	496
6 + 0 + 6 + 0	491
6 + 0 + 0 + 6	503
0 + 6 + 0 + 6	496
0 + 0 + 6 + 6	497
4+ 4 + 4 + 0	500
0 + 4 + 4 + 4	505
4 + 0 + 4 + 0	512

Final Remarks

- The first version of CLUSTERIX middleware is available
- More and more experiences with running application in CLUSTERIX environment
- CLUSTERIX is a promising platform for numerical computation, including meta-computations
- However, harnessing CLUSTERIX power by meta-applications needs to take into account hierarchical architecture of CLUSTERIX infrastructure, and its heterogeneity
- Extremely important for us:

  - to attract perspective users with new applications

  - to involve new dynamic clusters

  - training activities

  - . . . .

Thank YOU !

http://clusterix.pcz.pl

Roman Wyrzykowski
roman@icis.pcz.pl

Norbert Meyer
meyer@man.poznan.pl