



# Introduction to Grid and Grid applications



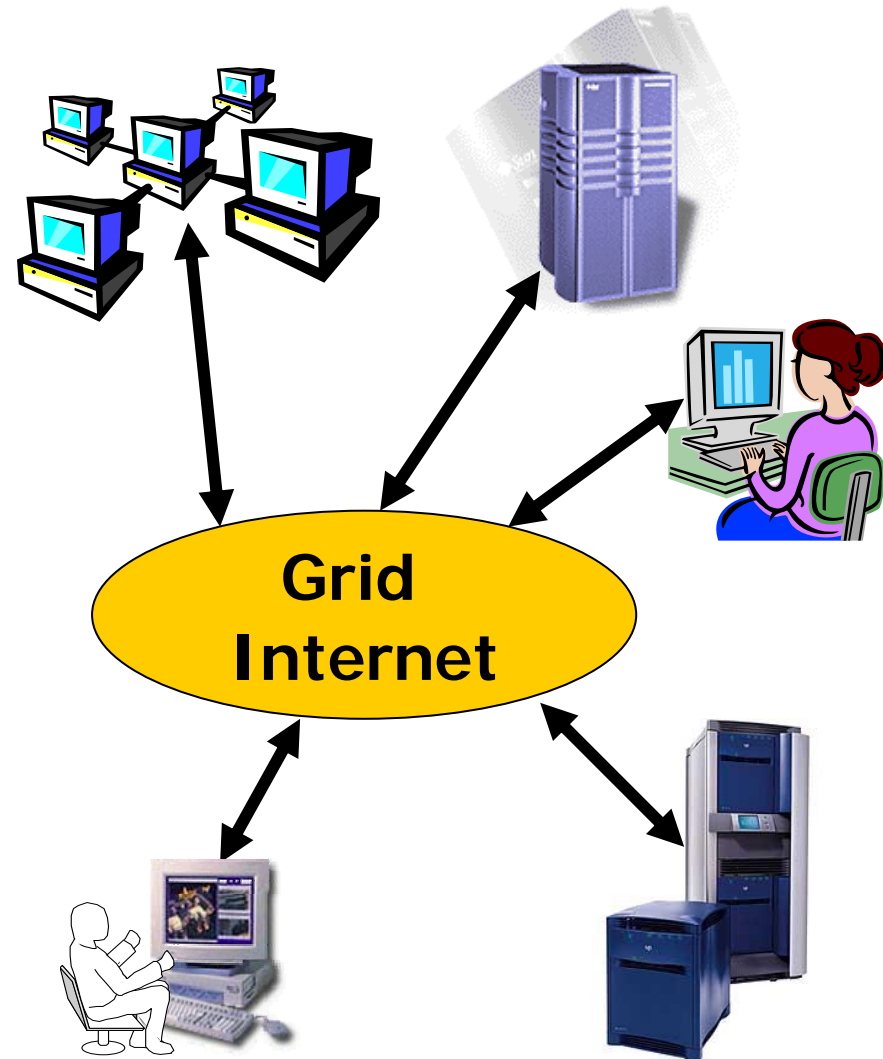
Peter Kacsuk and Gergely Sipos

MTA SZTAKI

[www.lpds.sztaki.hu](http://www.lpds.sztaki.hu)

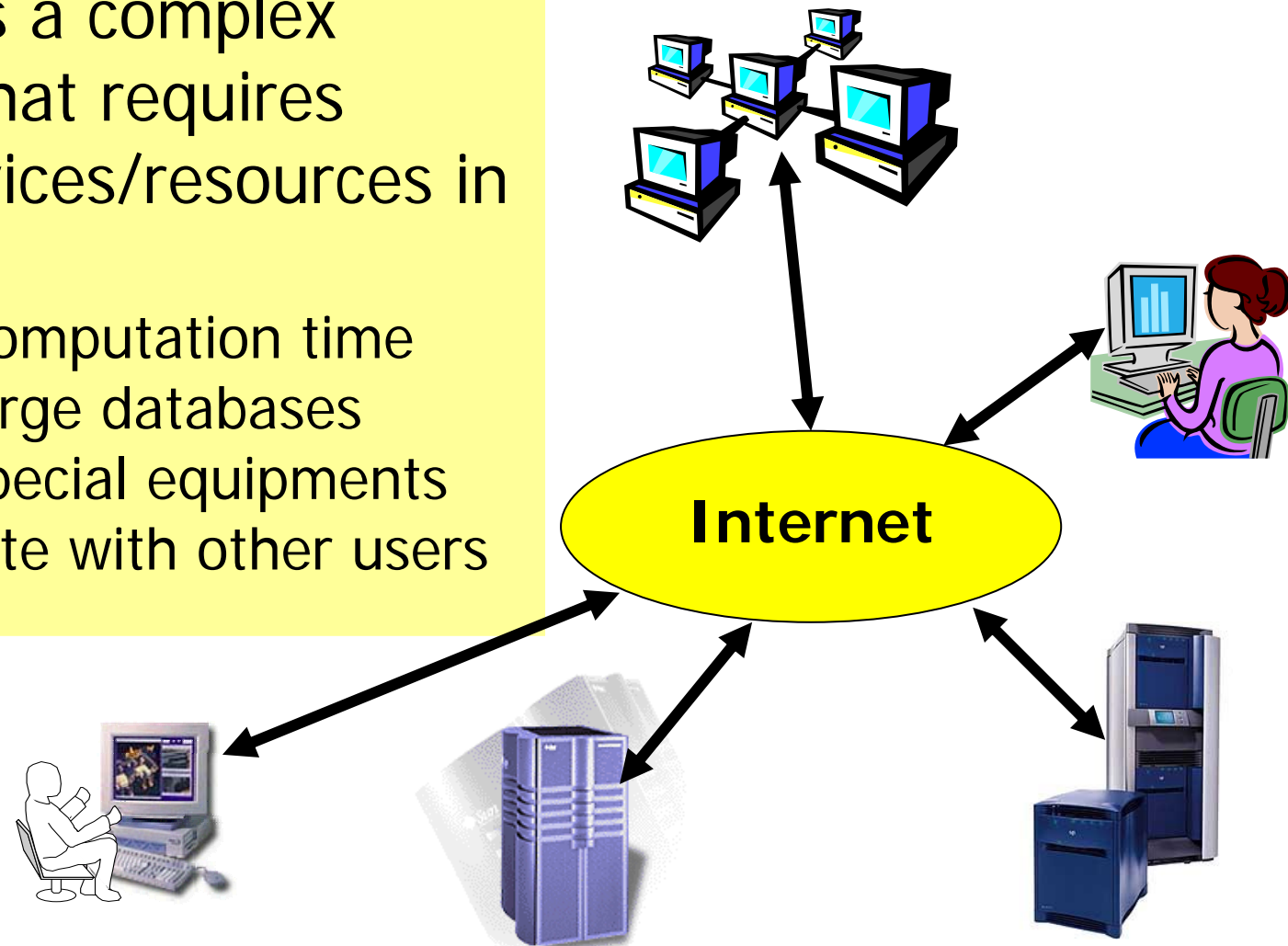
# What is Grid?

- A Grid is a collection of computers, storages, special devices, services that can **dynamically join and leave** the Grid
- They are **heterogeneous** in every aspect
- They are geographically **distributed** and connected by a **wide-area network**
- They can be accessed **on-demand** by a set of users



# Why use a Grid?

- A user has a complex problem that requires many services/resources in order to
  - reduce computation time
  - access large databases
  - access special equipments
  - collaborate with other users





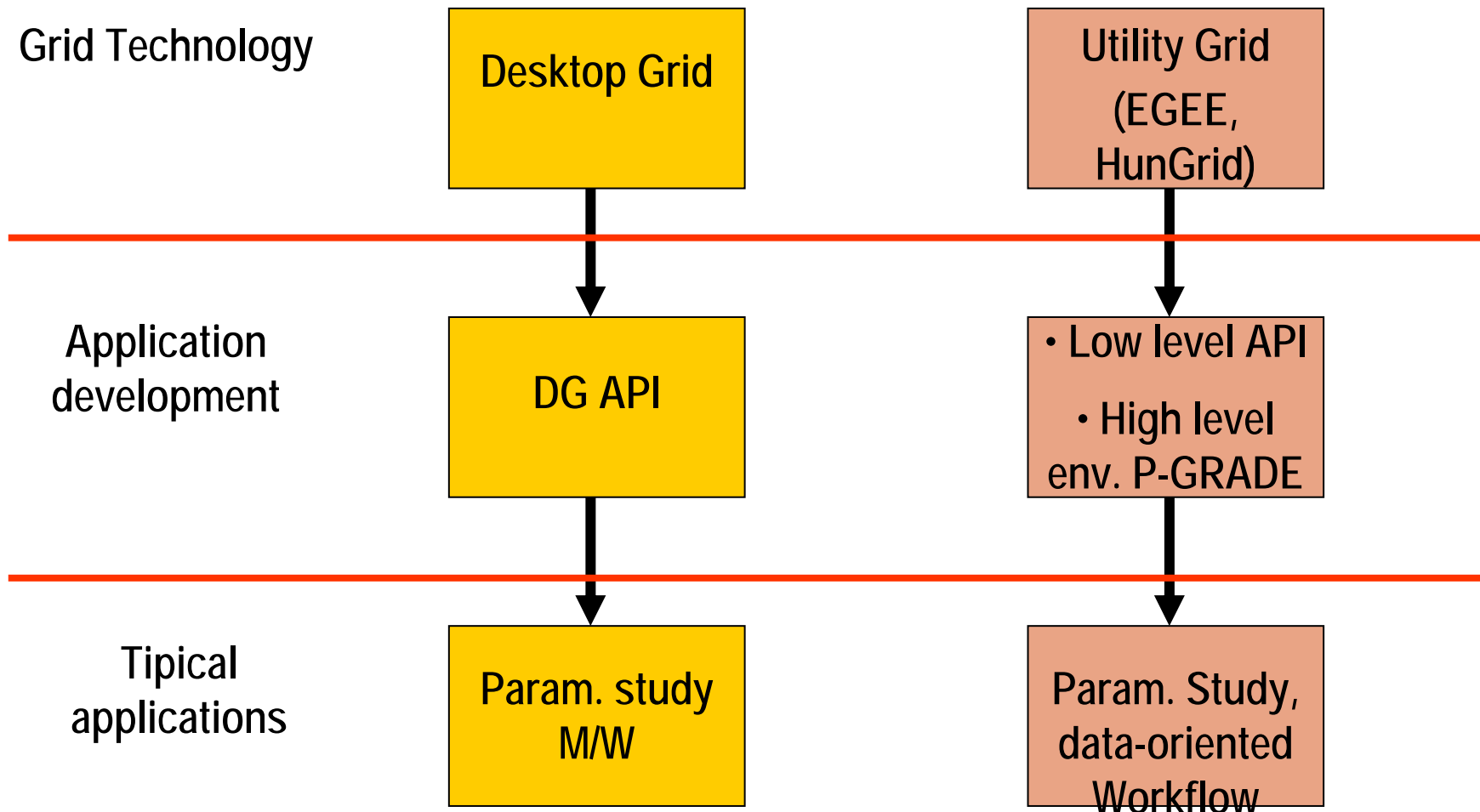
# Typical Grid application areas



- **High-performance computing (HPC)**
  - to achieve **higher performance** than individual supercomputers/clusters can provide
  - Requirement: **parallel computing**
- **High-throughput computing (HTC)**
  - To exploit the **spare cycles** of various computers connected by wide area networks
- **Collaborative work**
  - Several users can jointly and remotely solve complex problems



# Two basic Grid directions

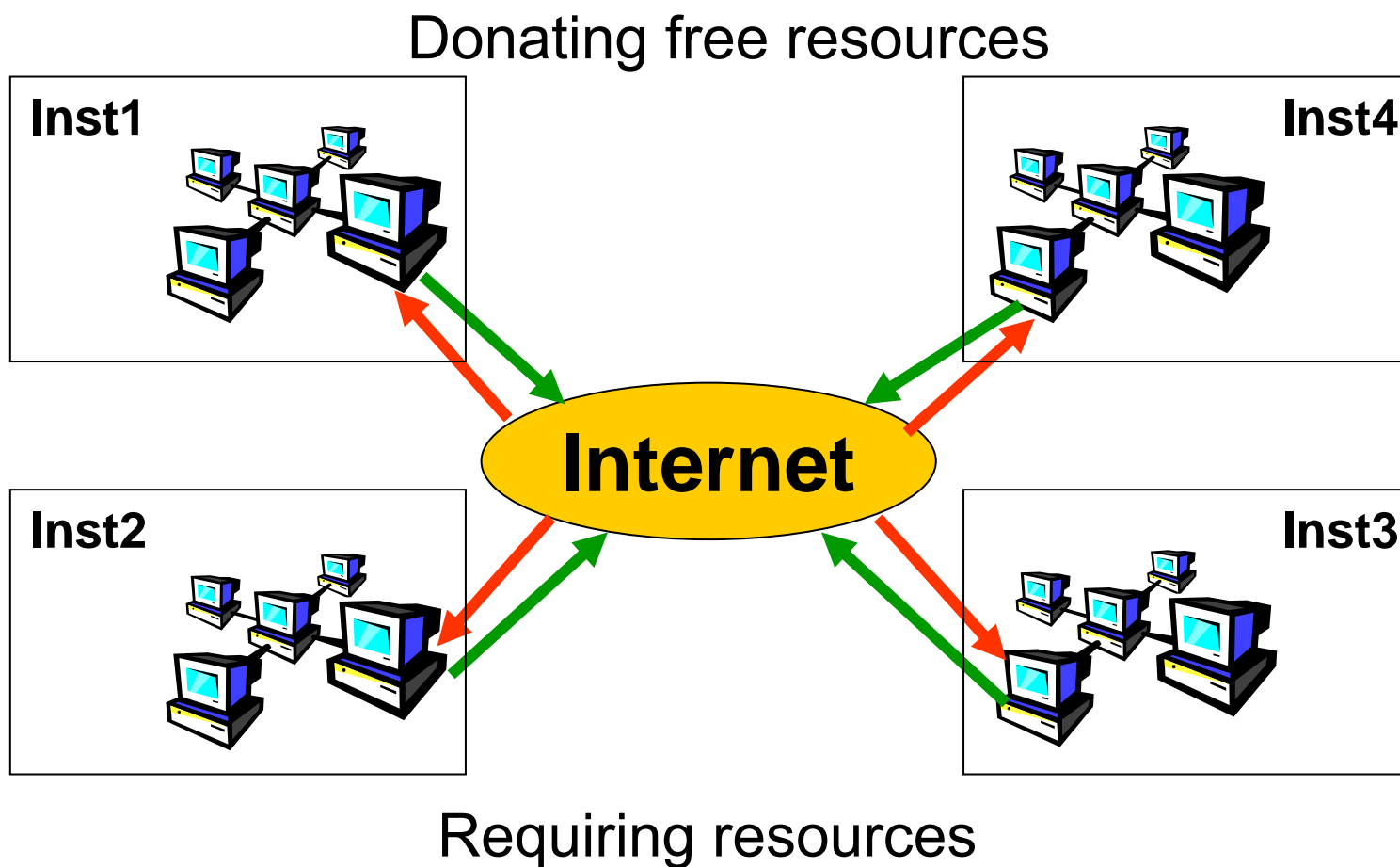




# Two players of the Grid



- Resource donors = D
- Resource users = U
- Relationship between the two characterizes the Grid:
  - if  $U \sim D$   $\Rightarrow$  generic Grid model
  - if  $U \gg D$   $\Rightarrow$  utility Grid model
  - if  $U \ll D$   $\Rightarrow$  desktop Grid model





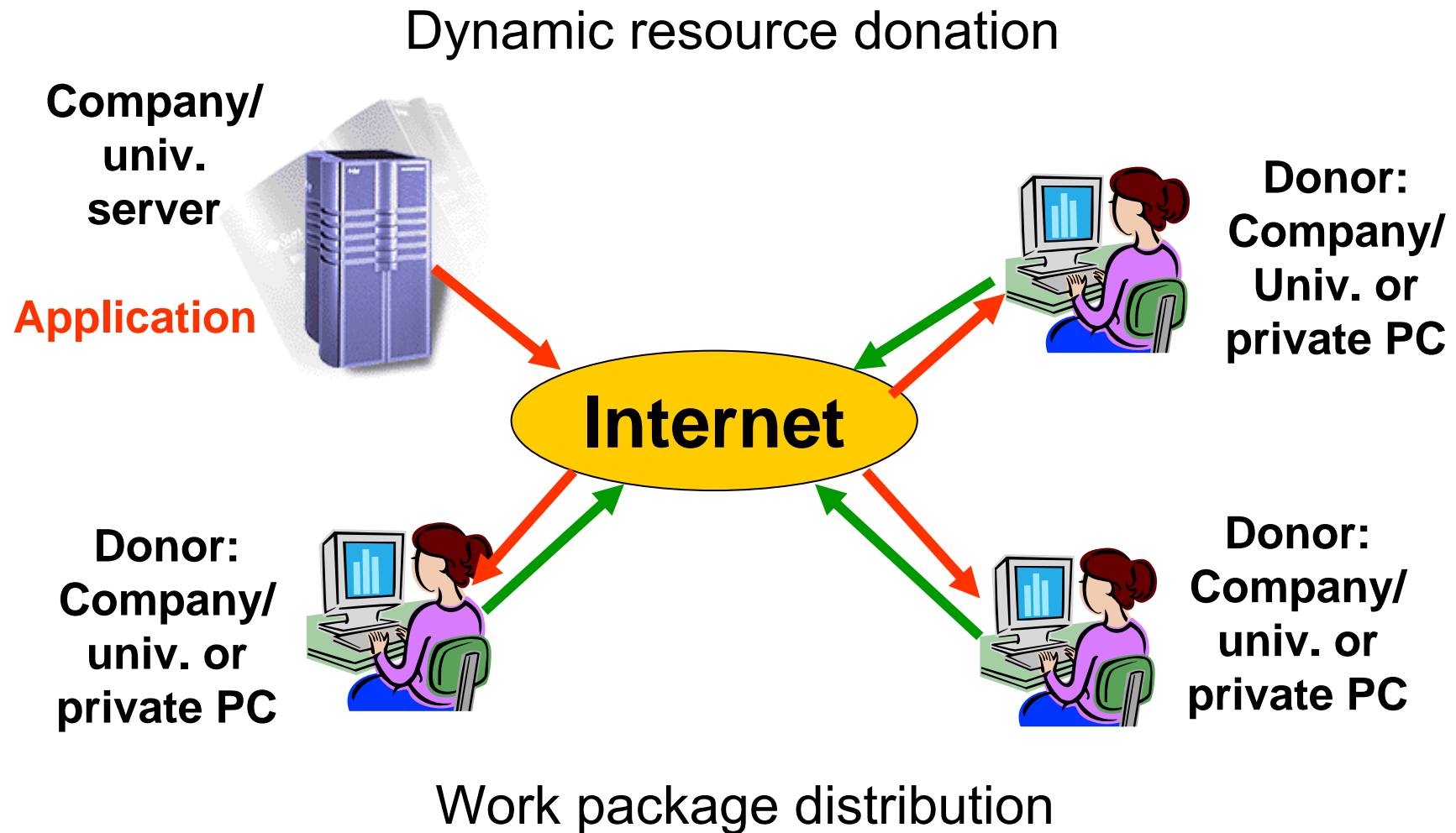
# Characteristics of the generic Grid model



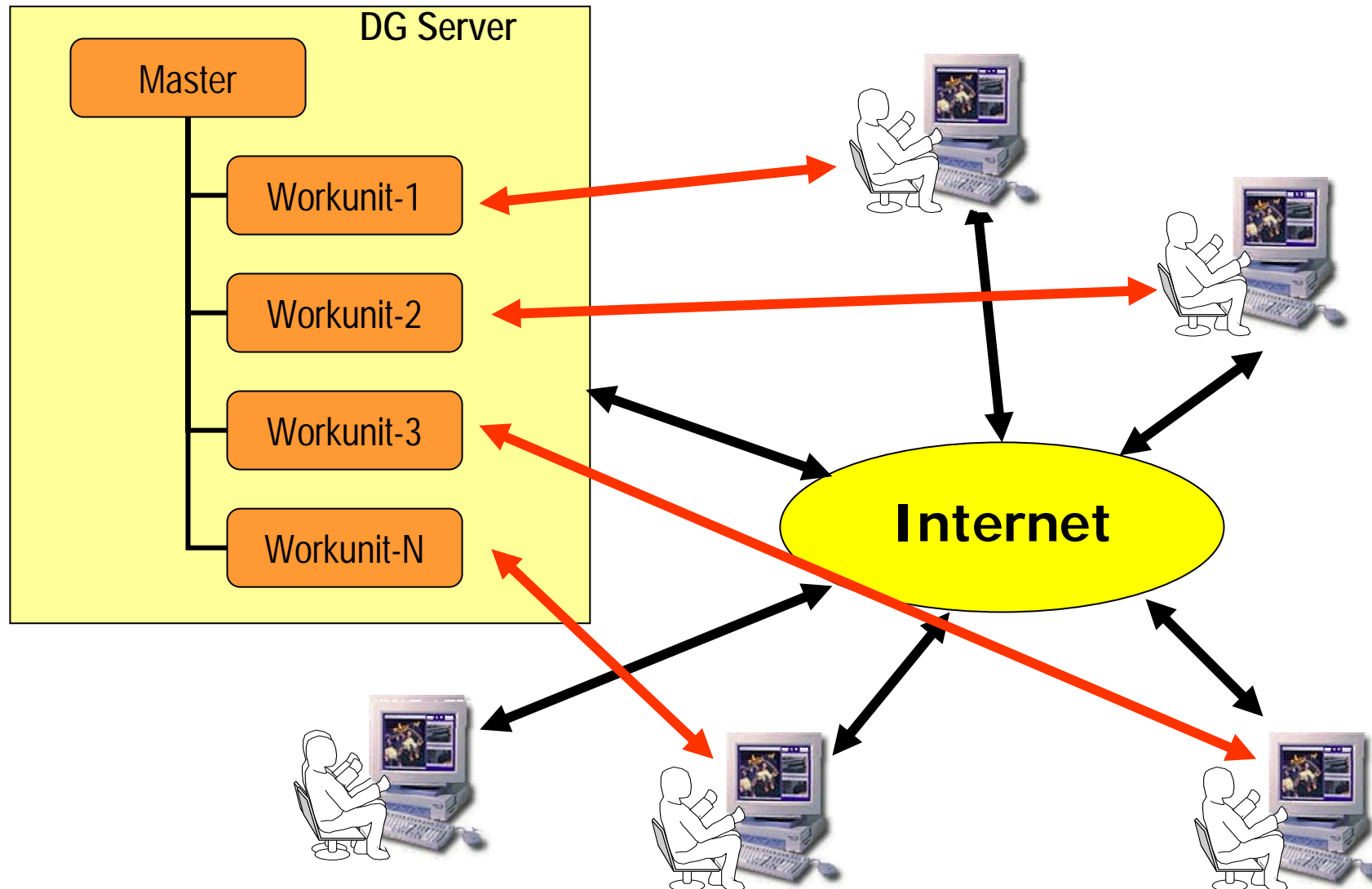
- Anybody can donate resources
- Heterogeneous resources, that dynamically join and leave
- Anybody can use the donated resources for solving her/his **own** applications
- Symmetric relationship between donors and users:  
$$U \sim D$$
- Examples:
  - GT-2 grids
  - Jini based JGrid (developed in Hungary)
- **Problems:** Installing and maintaining client and server grid software are too complicated



# Desktop Grid model



# Desktop Grid model – Master/slave parallelism





# Characteristics of the desktop Grid model



- Anybody can donate resources
- Heterogeneous resources, that dynamically join and leave
- **One or a small number of projects** can use the resources
- Asymmetric relationship between donors and users:  
$$U \ll D$$
- Advantage:
  - Donating a PC is extremely easy
  - Setting up and maintaining a DG server is much easier than installing the server sw of utility grids



# Types of Desktop Grids



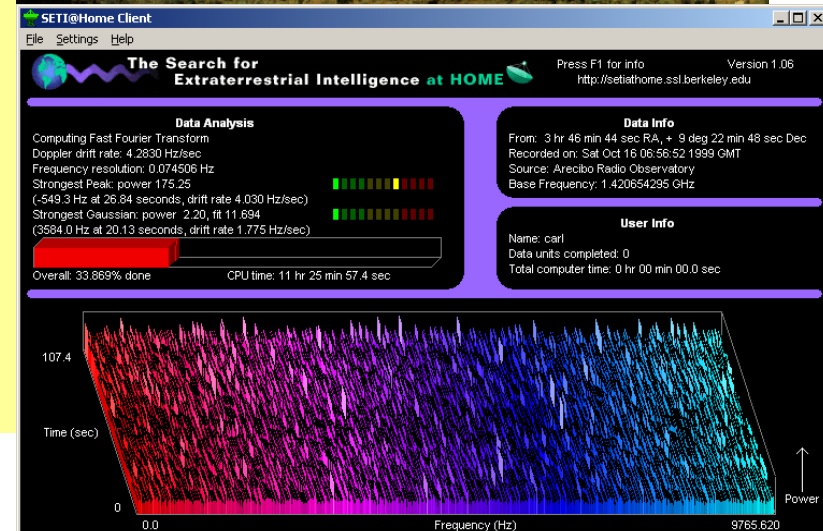
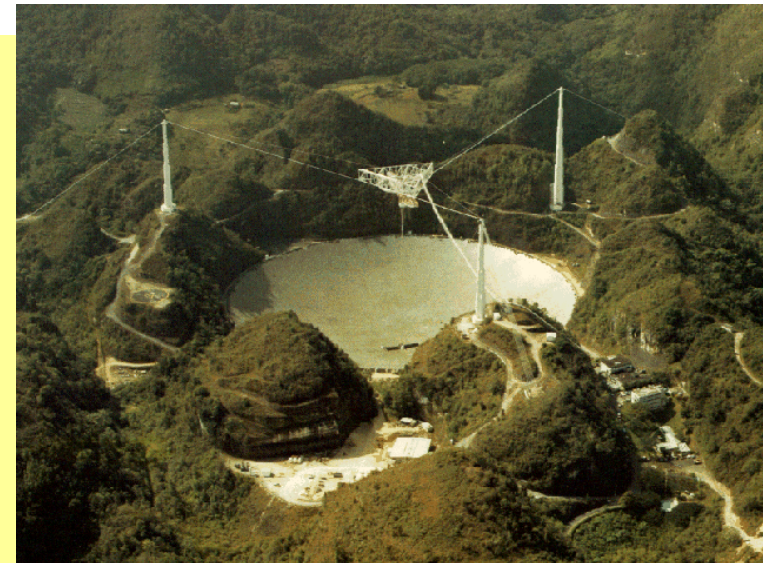
- Global Desktop Grid
  - Aim is to collect resources for grand-challenge scientific problems
- Example:
  - BOINC (SETI@home)
  - SZTAKI Desktop Grid
- Local Desktop Grid
  - Aim is to enable the quick and easy creation of grid for any community (company, univ. city, etc.) to solve their own applications
- Example:
  - SZTAKI Desktop Grid



# SETI: a global desktop grid



- SETI@home
  - 3.8M users in 226 countries
  - 1200 CPU years/day
  - 38 TF sustained (Japanese Earth Simulator is 32 TF sustained)
  - **Highly heterogeneous**: >77 *different* processor types






# SZTAKI Desktop Grid: Global version



- Main objective:
  - **Demonstrate the usage of DG systems for any community**
- Three steps to try and use the system:
  1. Donate one PC to test the client site on the SZDG
  2. Port application to the DG server and register PCs for that application
  3. Set up a DG server for the community
- SZTAKI helps in steps 2 and 3
- Number of registered donors: ~10000
- Number of registered PCs: > 15000
- How to register a PC?
  - <http://www.lpds.sztaki.hu/szdg/>



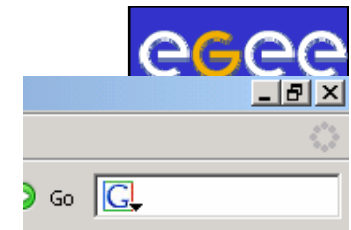
SZTAKI Desktop Grid - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://szd

Getting Started Latest Headlines

# SZTAKI Desktop Grid global version



eGee

Go





Project currently has **9984** registered users, with **15461** computers, the performance of which in the last 48 hours was **N/A** GFLOP/s. (More detailed diagrams are available [here](#))

**Join SZTAKI Desktop Grid**  

[Rules and policies \[read this first\]](#)  
[Getting started](#)  
[Frequently Asked Questions\(FAQ\)](#)  
[Create account](#)  
[Applications](#)

**Returning participants**


[Your account](#) - view stats, modify preferences  
[Teams](#) - create or join a team  
[Download BOINC client](#)


**Community**

**Application currently run by Project**

SZTAKI Desktop Grid currently searches for generalized binary number systems.  
 Description on the application is available [here](#).

**User of the day**



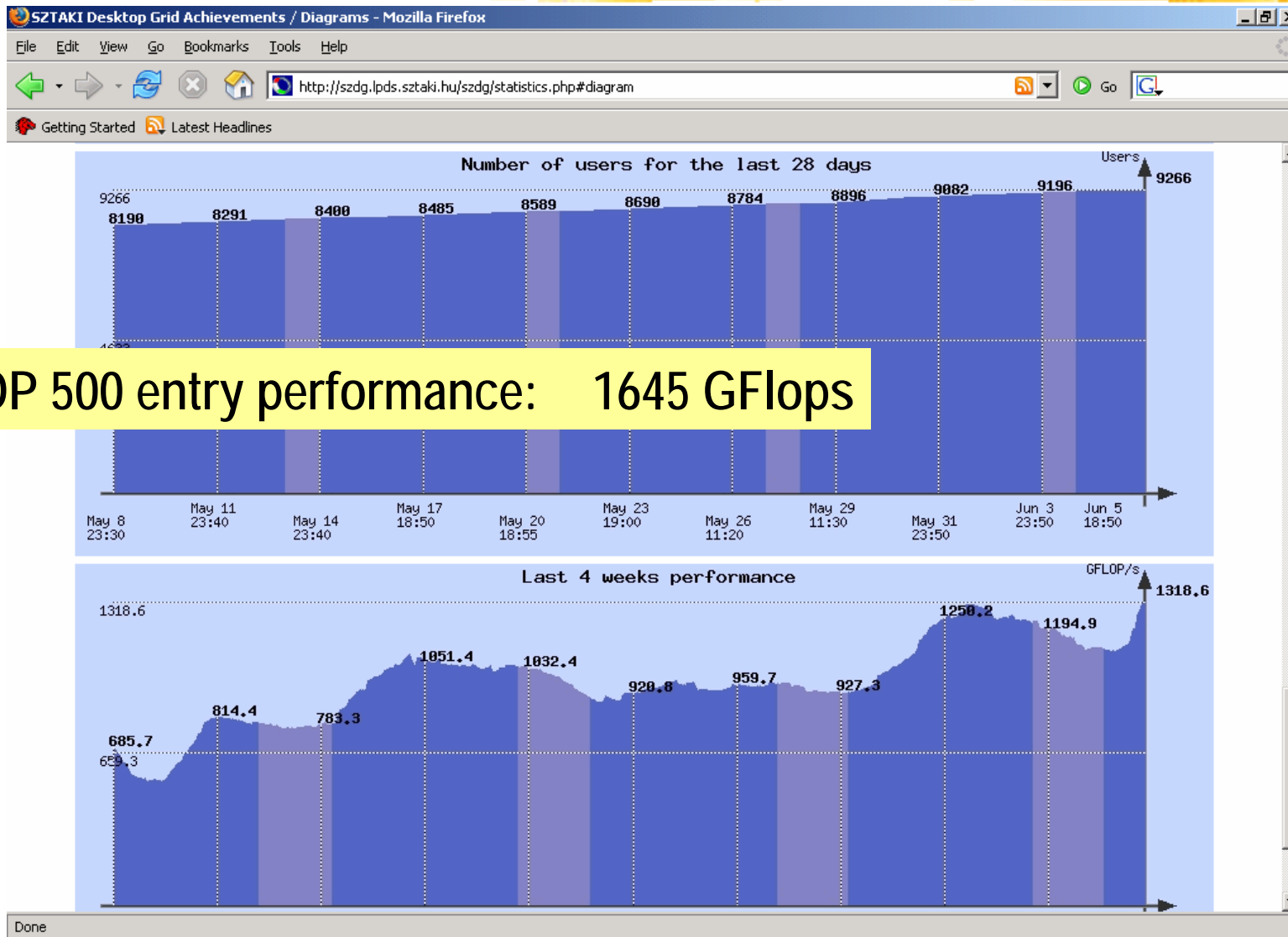
[ikarus1969](#) 

Hello! My name's Reinhart, i'm 36 years old and live in a small village near Vienna, Austria. Currently (2006) i am working as a programmer for





# SZTAKI Desktop Grid global version







# SZTAKI Desktop Grid local version



- **Main objective:**
  - **Enable the creation of local DG for any community**  
**Demonstrate how to create such a system**
  - Building production Grids requires huge effort and represents a privilege for those organizations where high Grid expertise is available
- **Using the local SZDG package**
  - **Any organization** can build a local DG in a day with minimal effort and with minimal cost (a strong PC is enough as a server machine)
  - The applications of the local community will be executed by the **spare PC cycles** of the local community
  - There is no limitation for the applied PCs, all the PCs of the organization can be exploited (**heterogeneous Grid**)

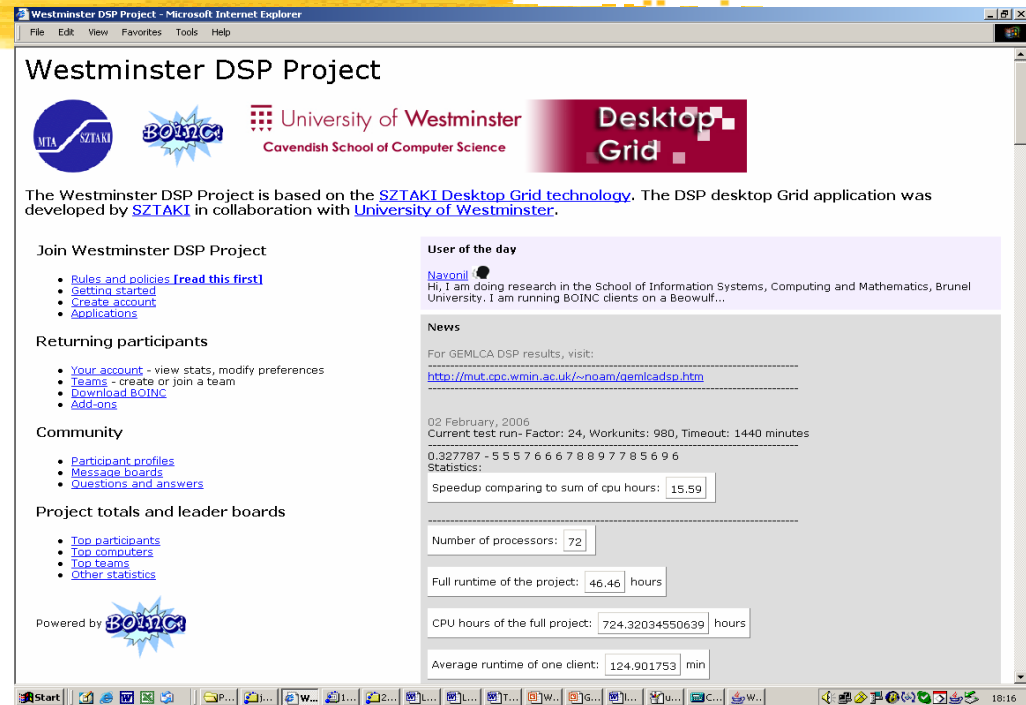


# DSP application on a local SZDG in the Univ. of Westminster



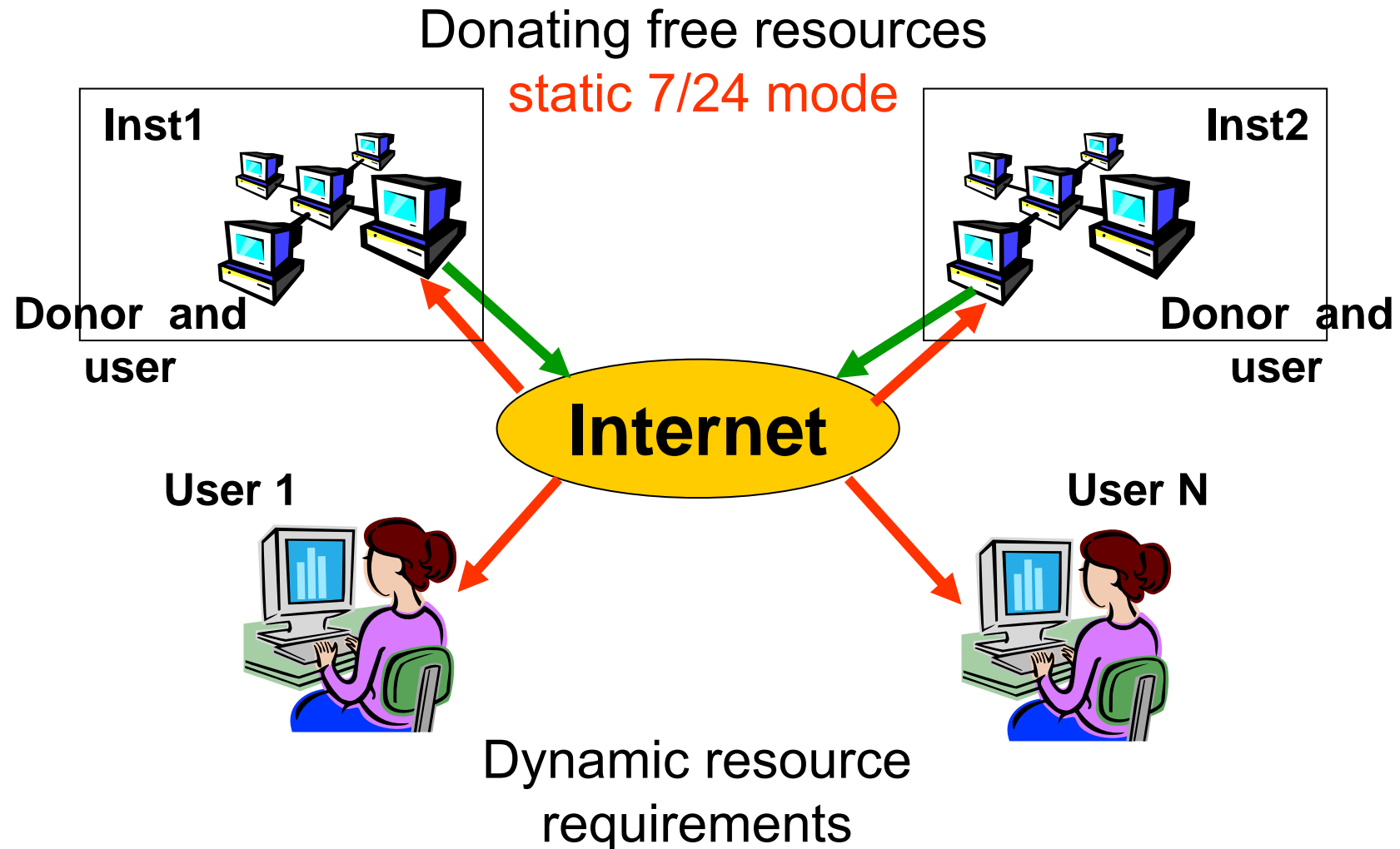
- **Digital Signal Processing Appl.:** Designing optimal periodic nonuniform sampling sequences
- Currently more than 100 PCs connected from Westminster and planned to extend over 1000 PCs

## The speedup



DSP size	Sequential	Production	SZDG
20	~3h 33min	~35min	~1h 44min
22	~41h 53min	~7h 23min	~5h 4min
24	~724h	~141h	~46h 46min

# Utility Grid model





# Characteristics of the utility Grid model

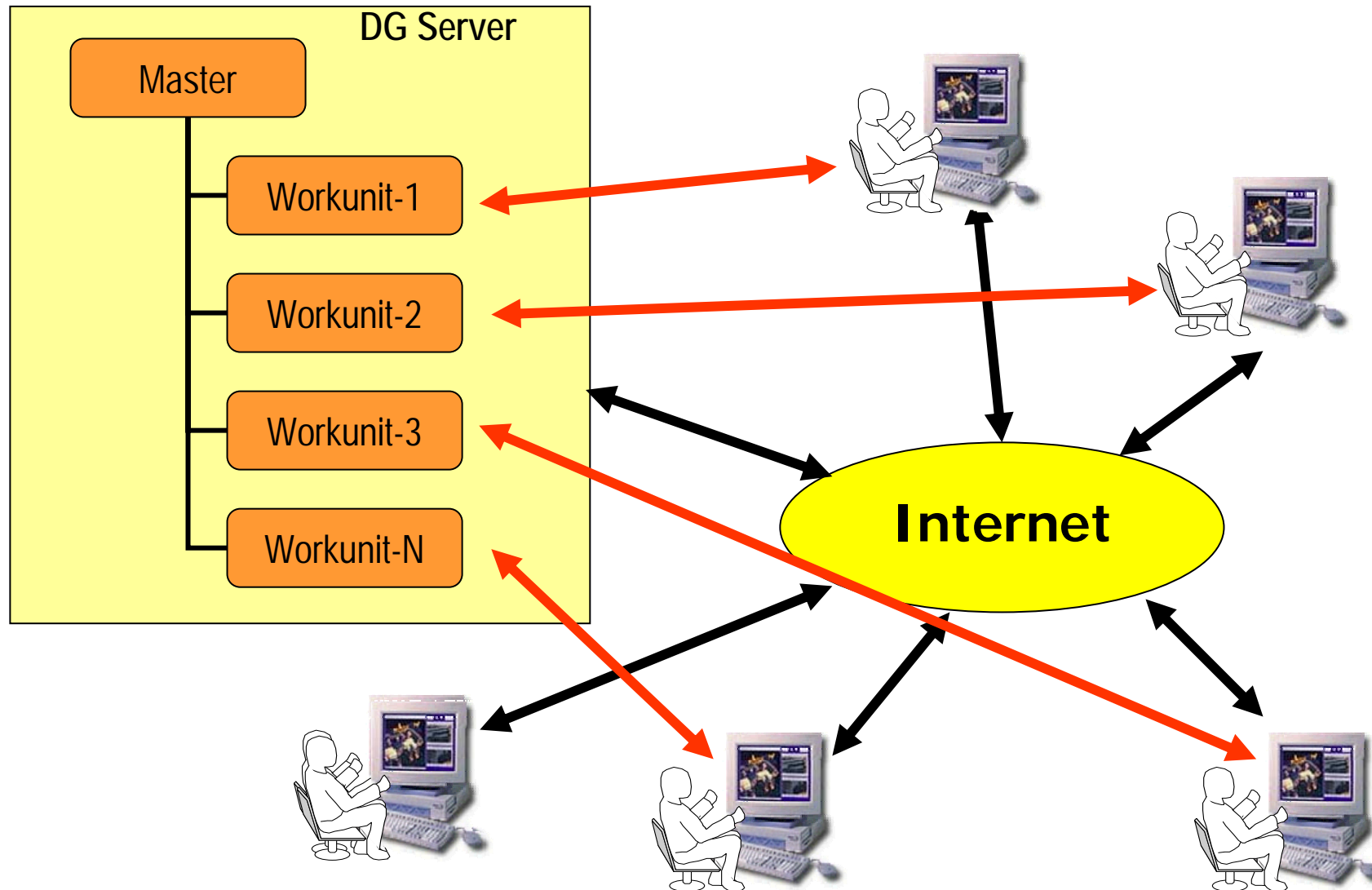


- Donors can be only “professional” resource providers who provide production service (7/24 mode)
- Homogeneous resources
- Anybody can use the donated resources for solving her/his **own** applications
- Asymmetric relationship between donors and users:

$$U \gg D$$

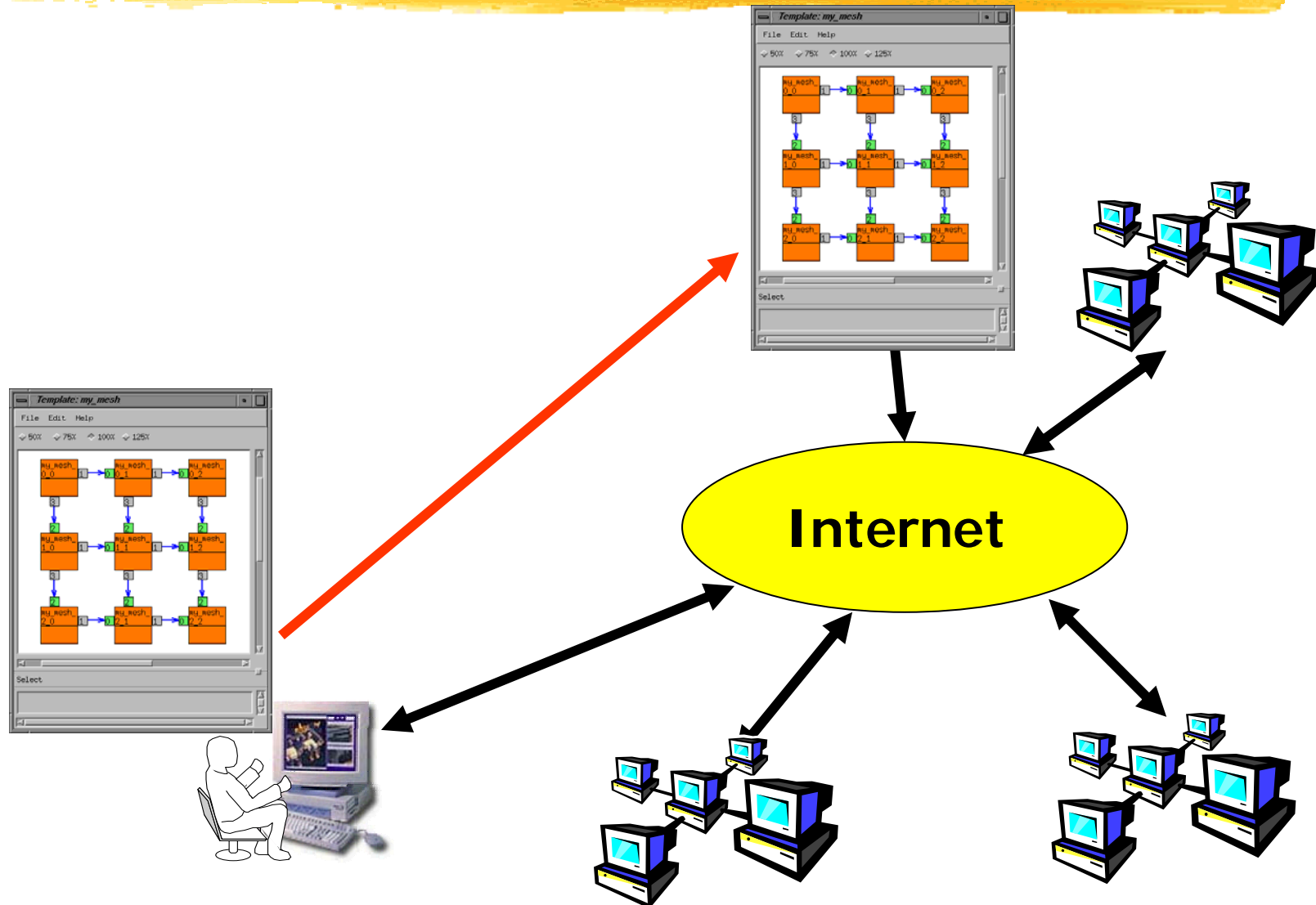
- Examples:
  - **EGEE -> HunGrid, SEE-Grid**
  - UK NGS, NorduGrid (Europe)
  - Grid2003, TeraGrid (USA)

# Master/slave parallelism and parametric studies in utility Grids



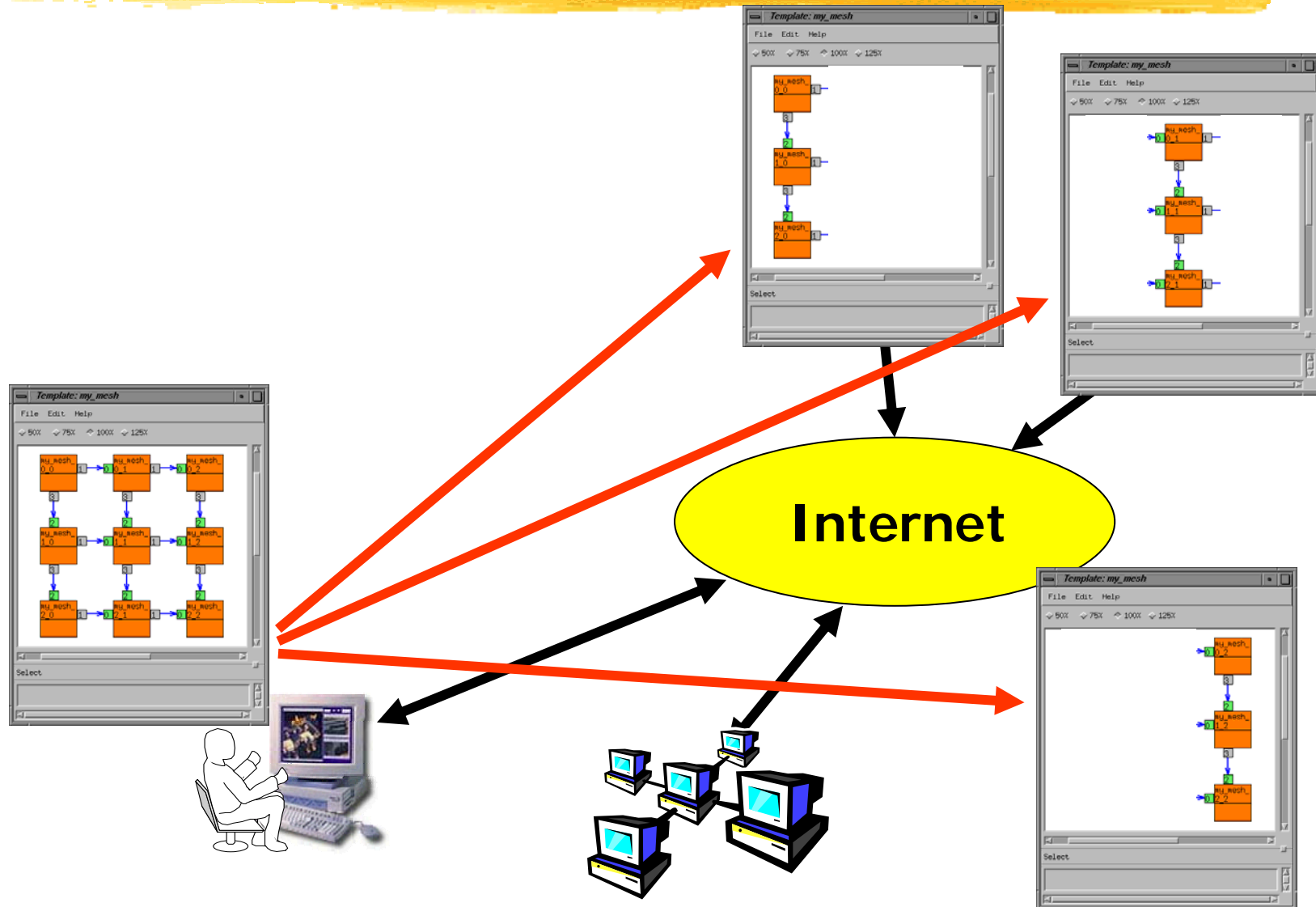


# How to use a Grid for single-site parallelism?

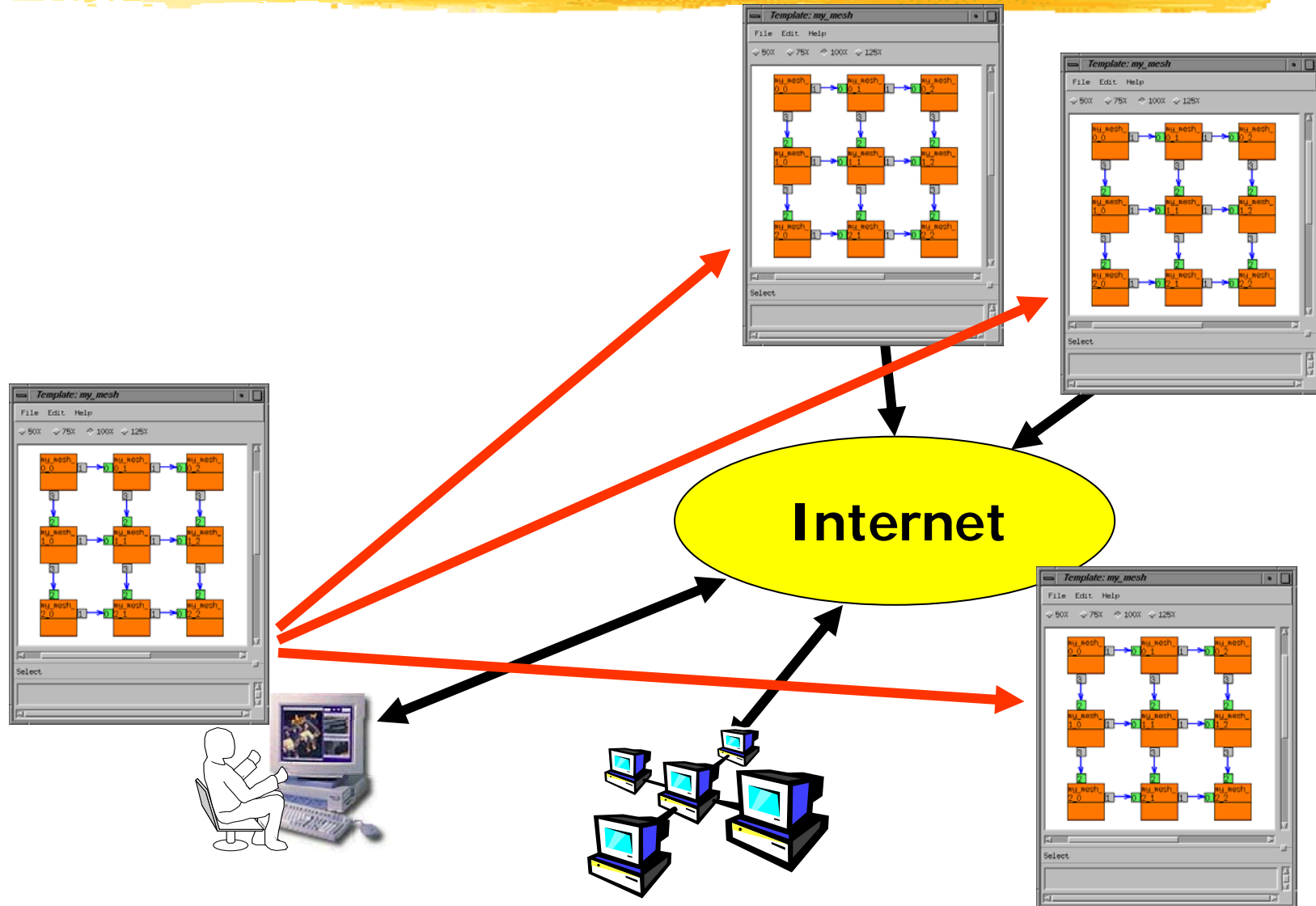




# How to use a Grid for multi-site parallelism?

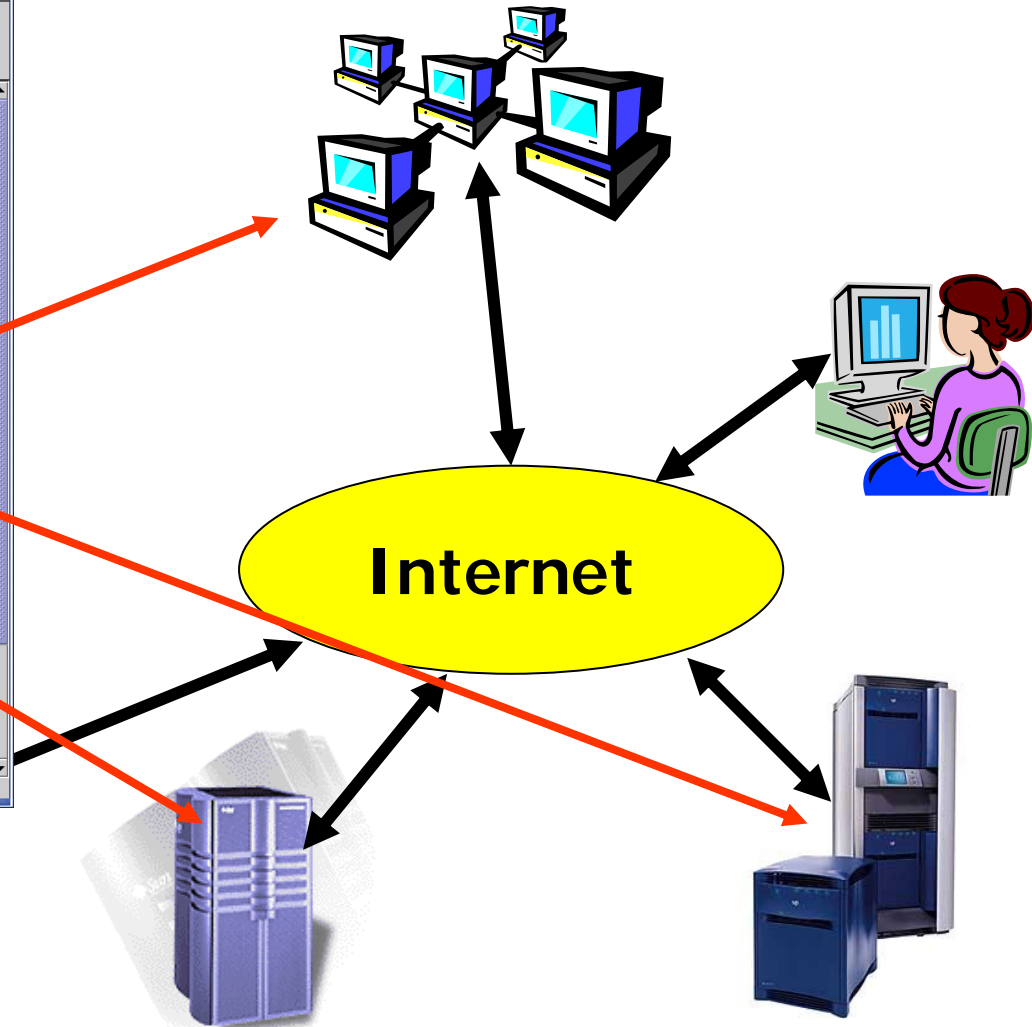
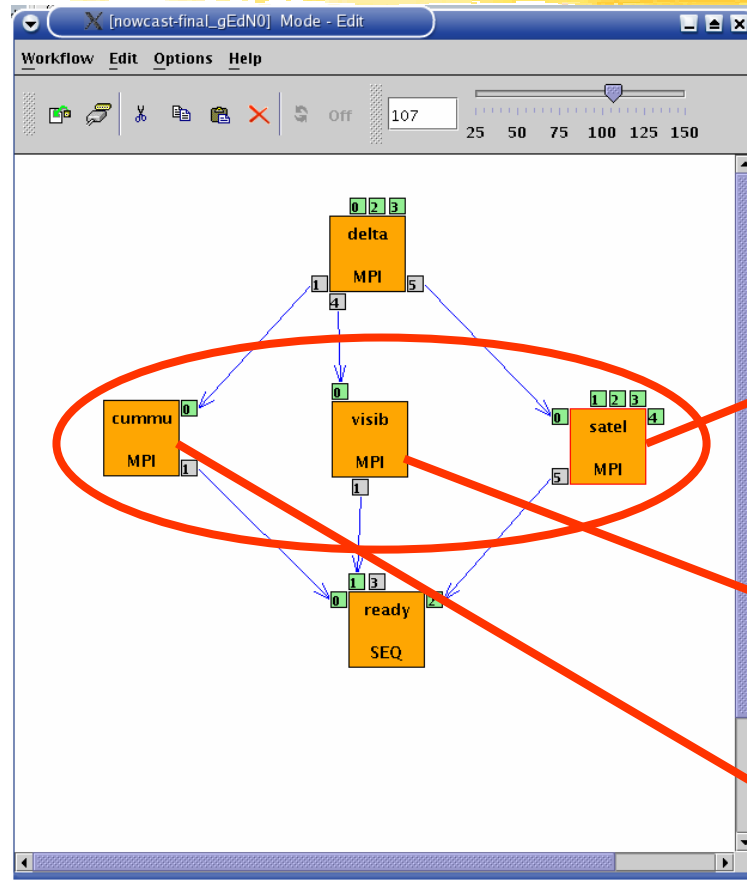


# How to use a Grid for HPC parameter study?



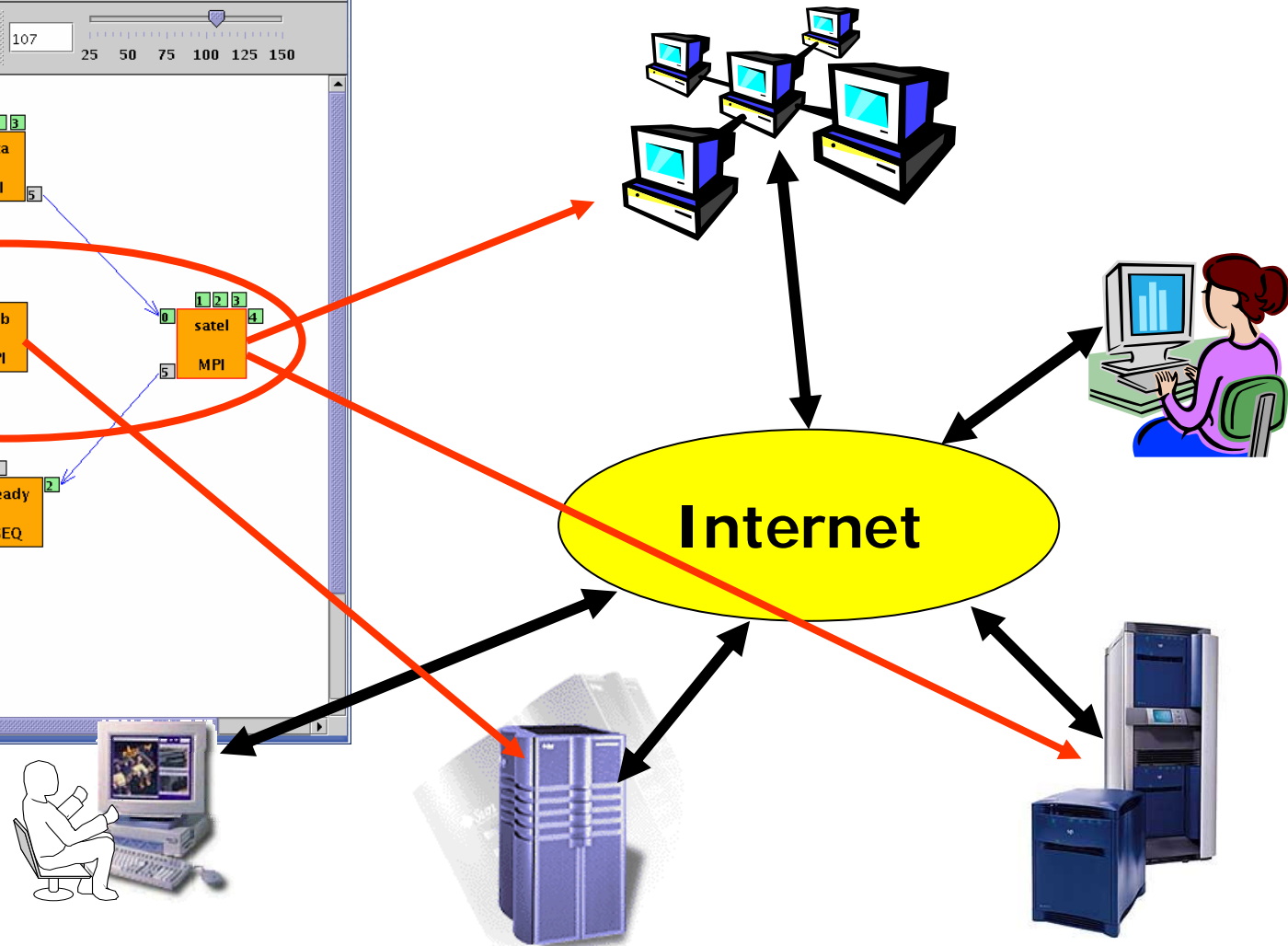
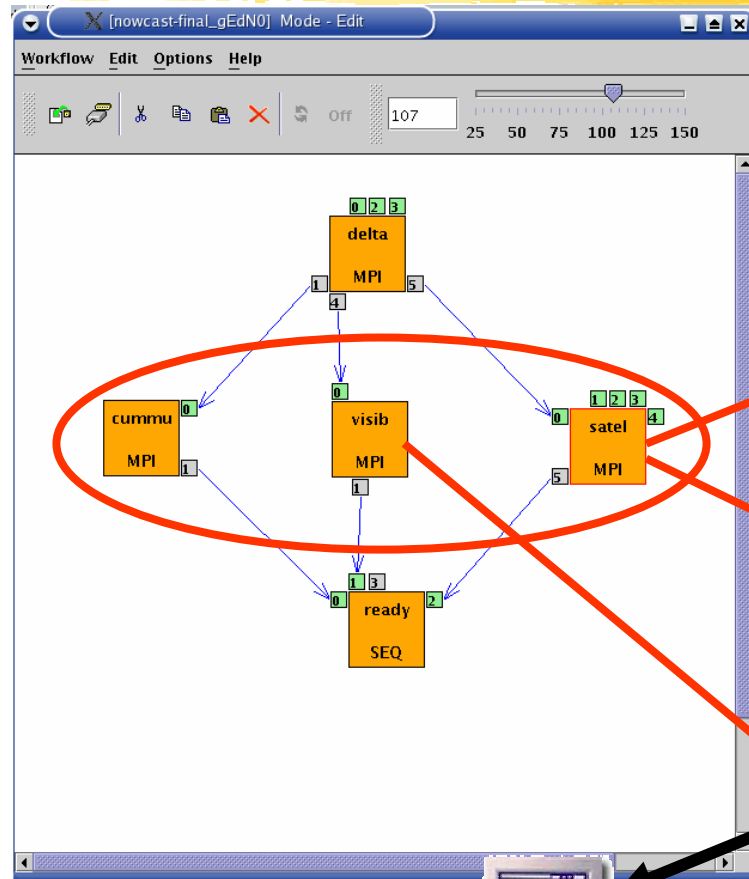


# How to use a Grid for two level single-site parallelism?





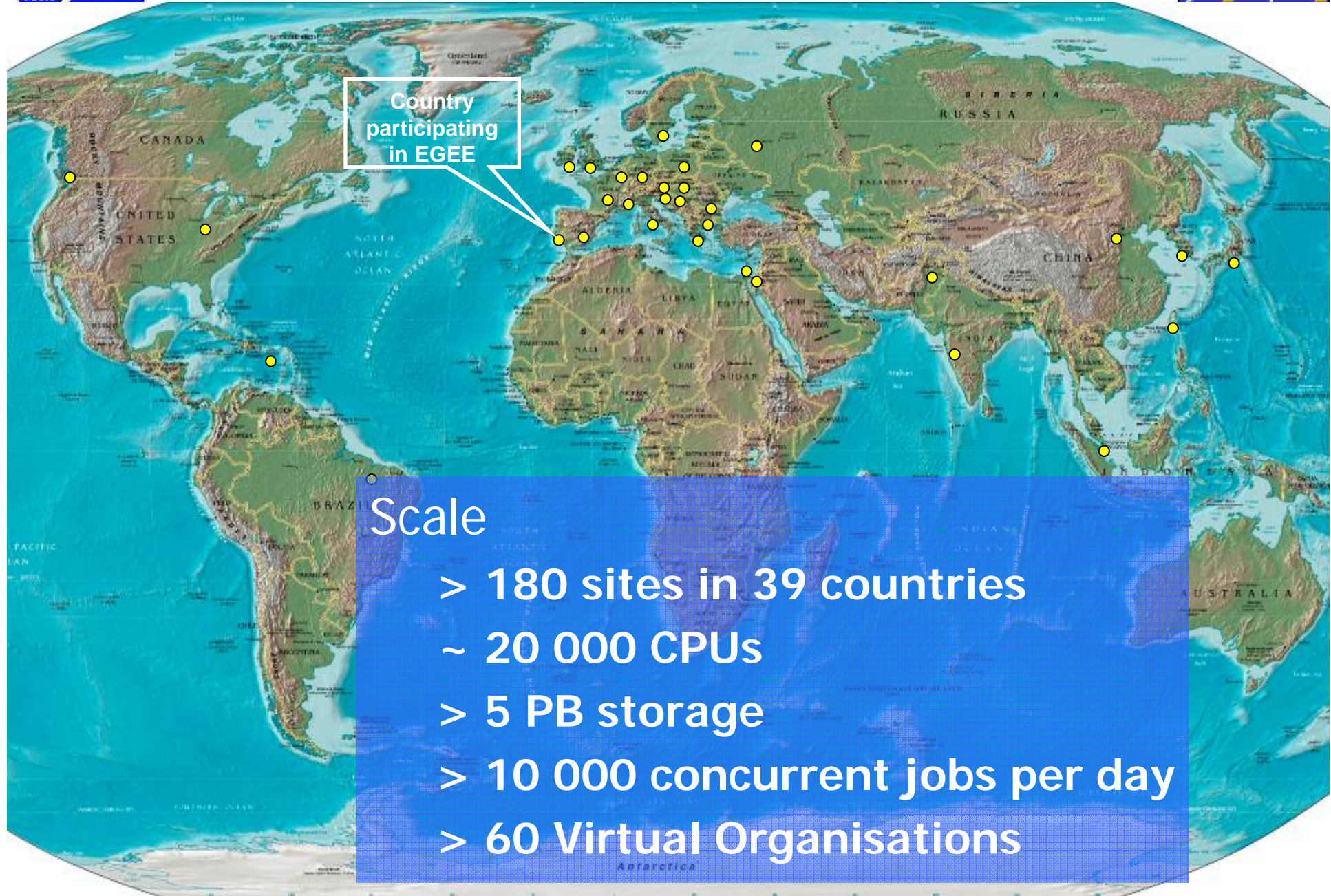
# How to use a Grid for two level multi-site parallelism?







# The largest production Grid: EGE







# NorduGrid



Dynamic Grid

~ 33 sites, ~1400 CPUS

Production Grid

Real users, real applications

It is in 24/7 operation, unattended by administrators for most of the time



# TeraGrid



## Caltech: Data collection analysis

0.4 TF IA-64  
IA32 Dataw...  
80 TB Storage



## LEGEND



Cluster



Visualization Cluster

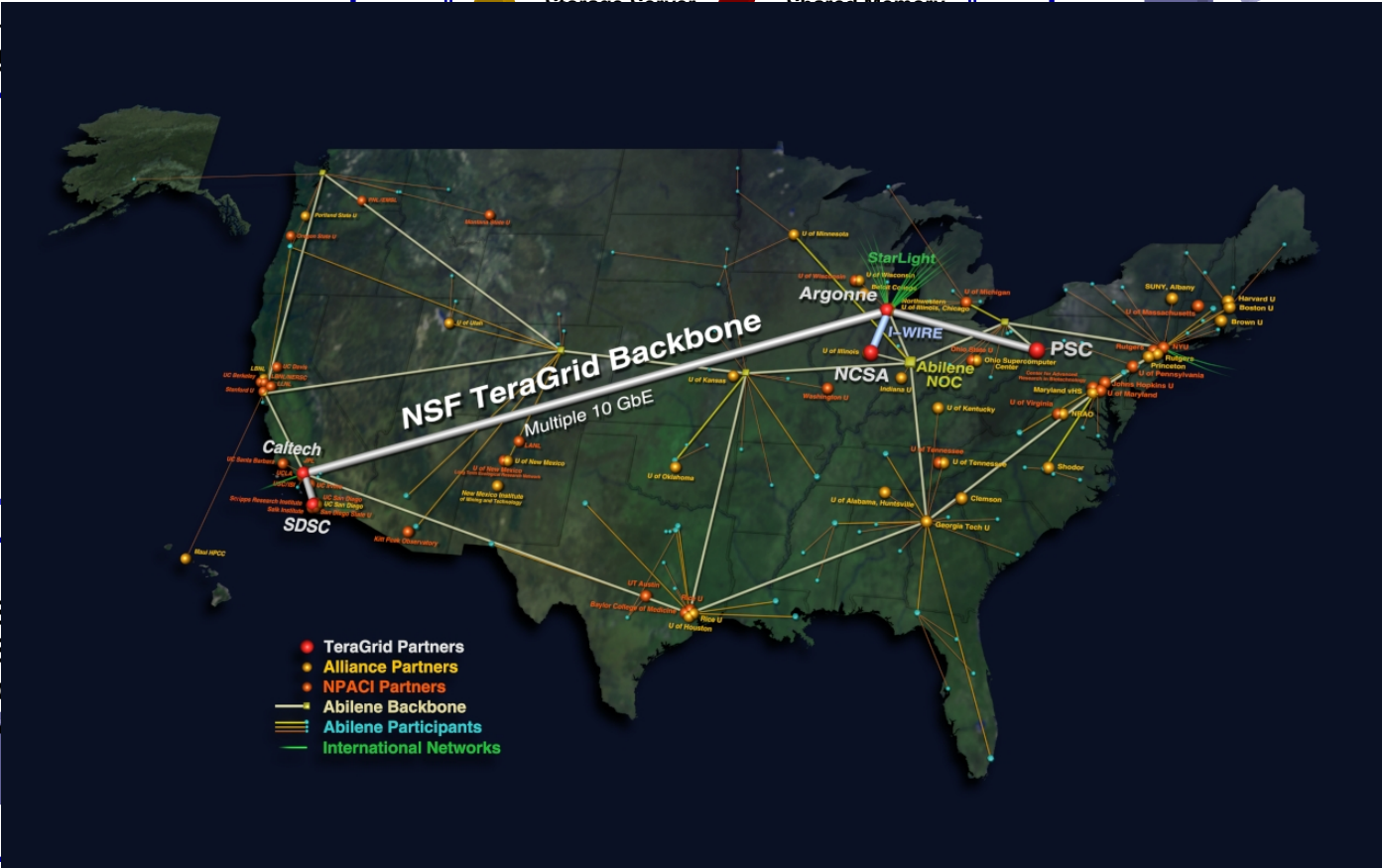
## ANL: Visualization



1.25 TF IA-64  
96 Viz nodes  
20 TB Storage

4 TF IA-64  
DB2, Oracle S...  
500 TB Disk S...  
6 PB Tape St...  
1.1 TF Power

SDSC



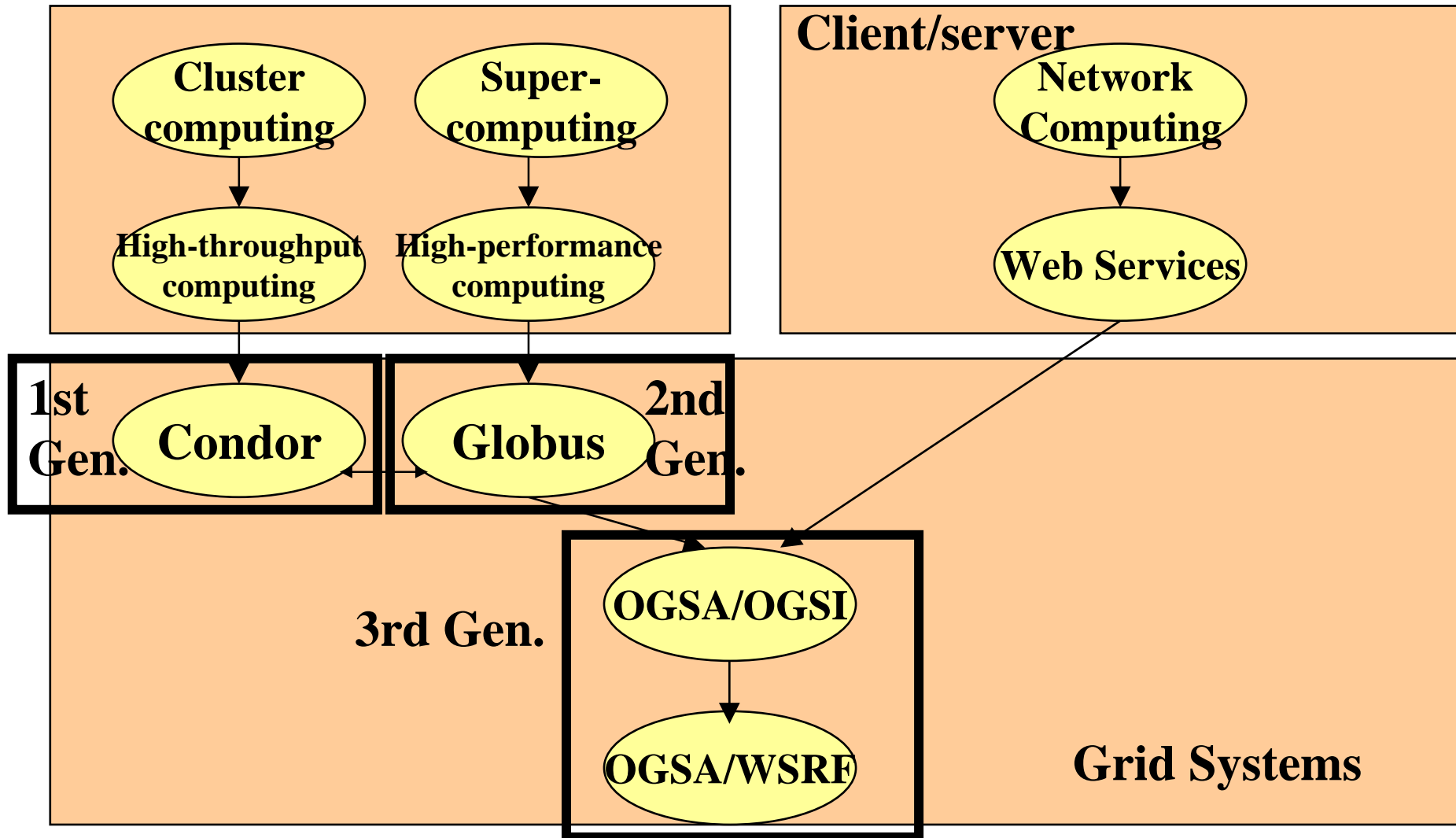
6 TF EV68  
71 TB Storage  
Shared-memory  
Storage Server

Intensive

PSC integrated Q3 03



# Progress in Grid Systems





# User concerns of Grid systems



- How to cope with the variety of these Grid systems?
- How to develop/create new Grid applications?
- How to execute Grid applications?
- How to observe the application execution in the Grid?
- How to tackle performance issues?
- How to port legacy applications
  - to Grid systems
  - between Grid systems?
- How to execute Grid applications over several Grids in a transparent way?



# Goal of the Grid Computing School



- This is a **user-oriented** and not a grid constructor **school** with goals:
  - To give answers for the questions above
  - Concentrating on the EGEE Grid technology but showing other generic solutions, too
  - Teaching the low-level EGEE user interface and APIs in an introductory level
  - Showing high-level Grid portal interfaces
  - Showing how to develop workflow-oriented Grid applications for the EGEE Grid and for other Grids