Grids: Why, How, and What Next

J. Templon, NIKHEF

ESA Grid Meeting

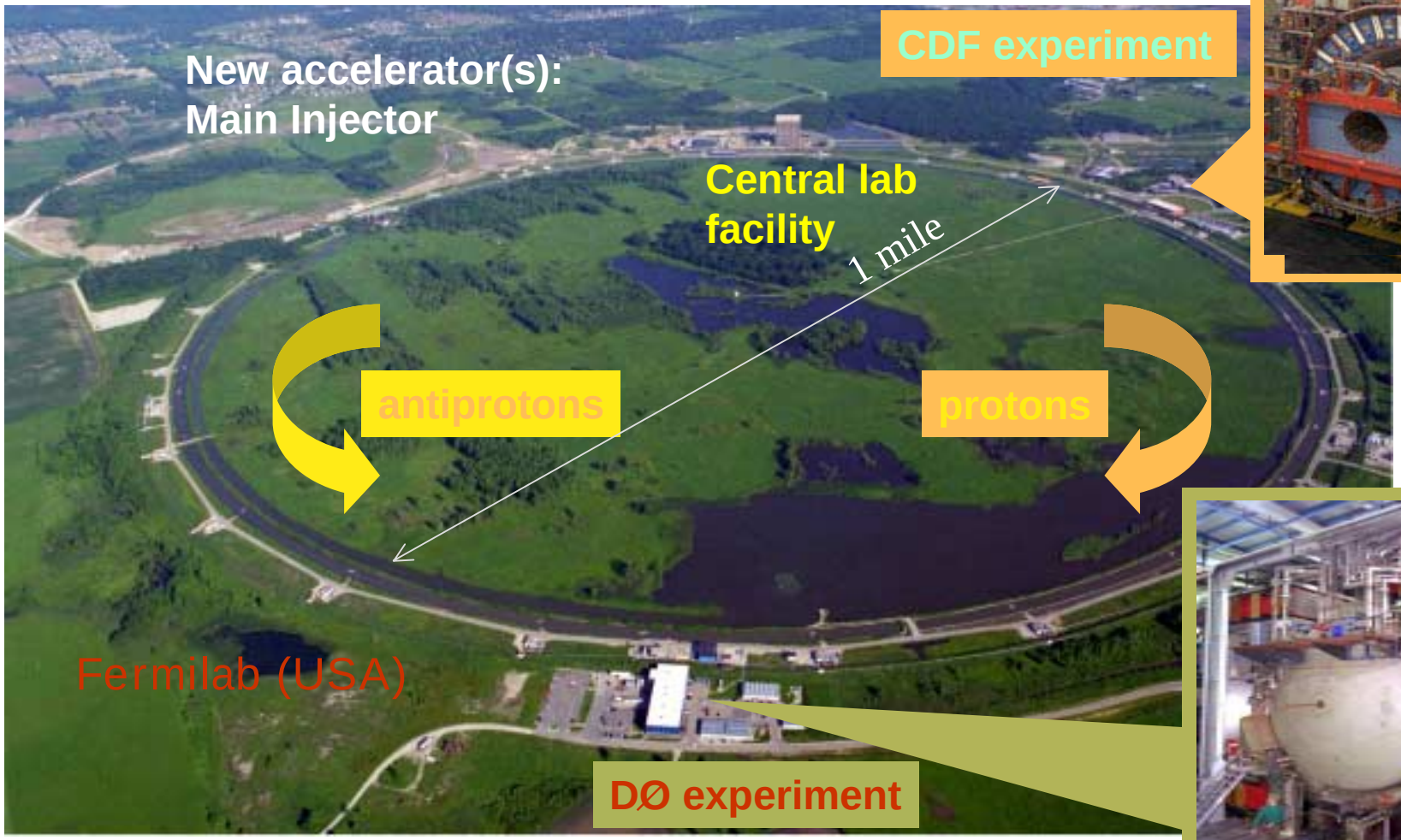
Noordwijk

25 October 2002

Information I intend to transfer

- Why are Grids interesting? Grids are *solutions* so I will spend some time talking about the *problem* and show how Grids are relevant. ***Solutions should solve a problem.***
- How are we (high-energy physicists) using Grids? What tools are available?
- What's next? In particular:
  - Short-term (next 12 months) plans of the European DataGrid project
  - Longer-term needs of the HEP community
  - Emerging trends in Grid computing – what should we watch closely for the next couple years?

High-Energy Physics



Why Collide Protons & Antiprotons?

- Look for particles – most interesting phenomena come with carrier particles
  - Photoelectric effect (& solar cells) – *photons*
  - Nuclear fusion – *pions* and other mesons
  - Radioactive decay – *W* and *Z* particles
- These particles are active within nuclei (like protons or antiprotons) but we want to take them out and study them. Sometimes we see the phenomenon, but we don't know how it works – finding the carrier particle helps a lot!
- Analogy: suppose cars occurred in nature, but were made so that you couldn't take them apart (*e.g.* with screwdrivers and wrenches) and you couldn't look inside

How to study "sealed cars"

- Collide them at high speed into a wall!
  - Look at the fragments
  - In some collisions, motor will fly out – cars have motors!
- Can't take motor apart – need higher speeds!
- Some brilliant soul realizes that high-speed, head-on collisions of two cars results in even more fragments
- In high-energy physics, we're colliding our "cars" (protons) in order find out how the spark plugs work
- At the LHC (CERN) we want to discover the particle responsible for how things in the universe have mass

CERN

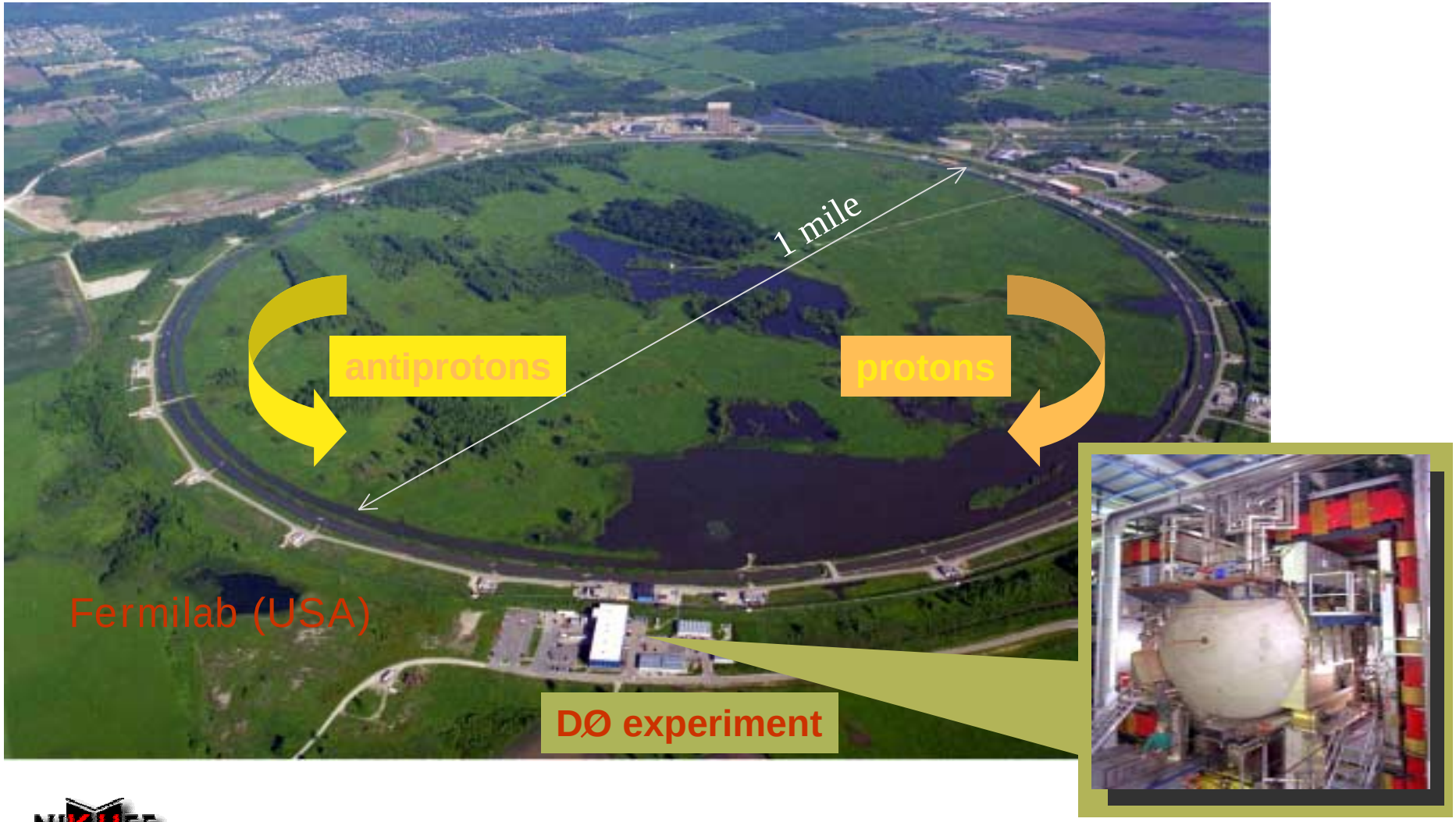
The European Organisation for Nuclear Research

20 European countries

2,700 staff

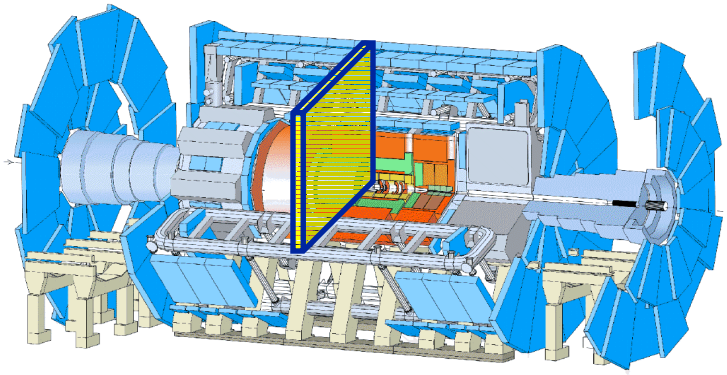
6,000 users

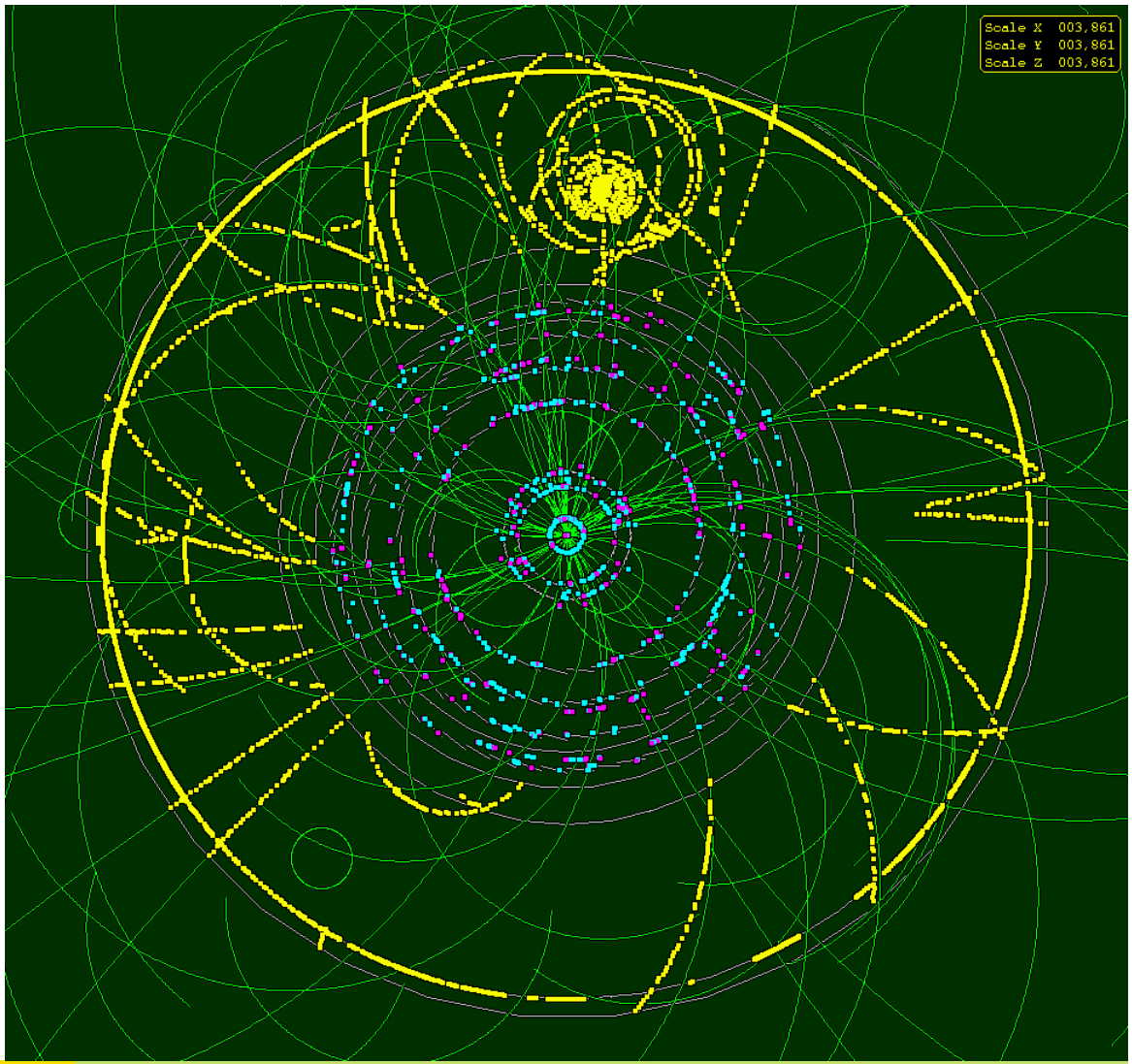
Detecting the Fragments



Detecting the Fragments (2) the DO detector at Fermilab

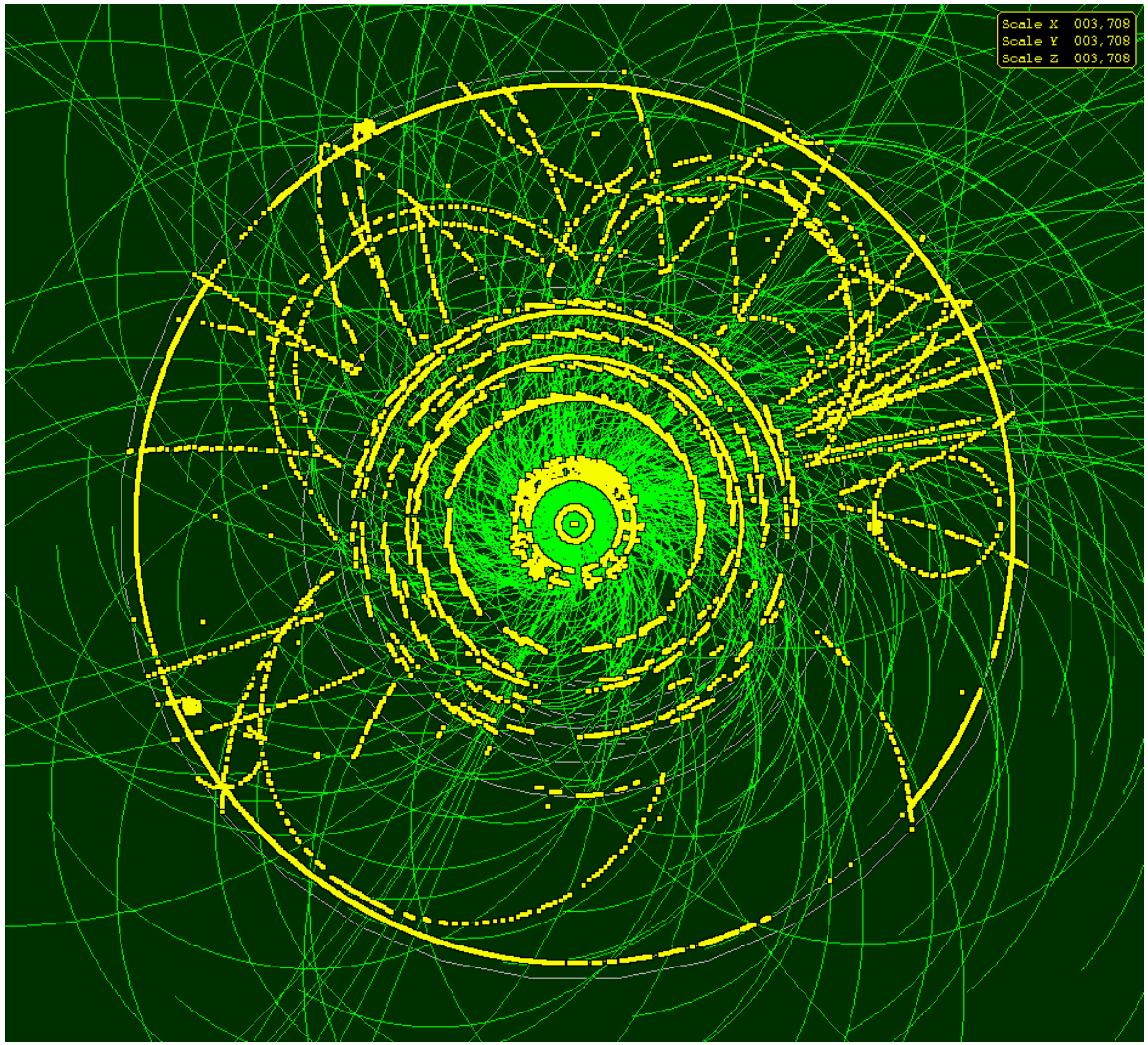
What do collisions look like?





- Place event info on 3D map
- Trace trajectories through hits
- [still needs work!]
- Assign type to each track
- Find particles you want
- Needle in a haystack!
- This is “relatively easy” case

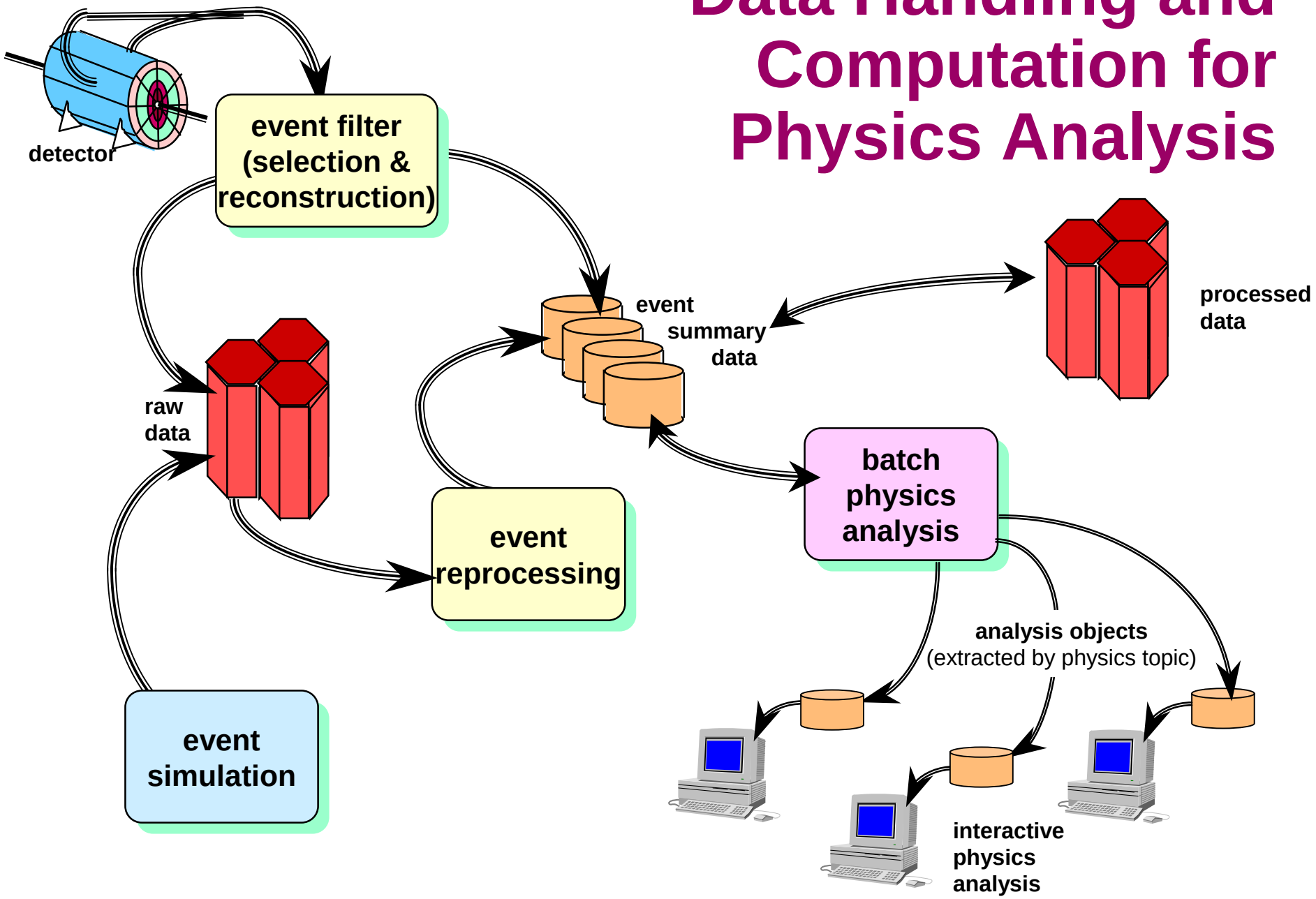
More complex example



Computational Implications

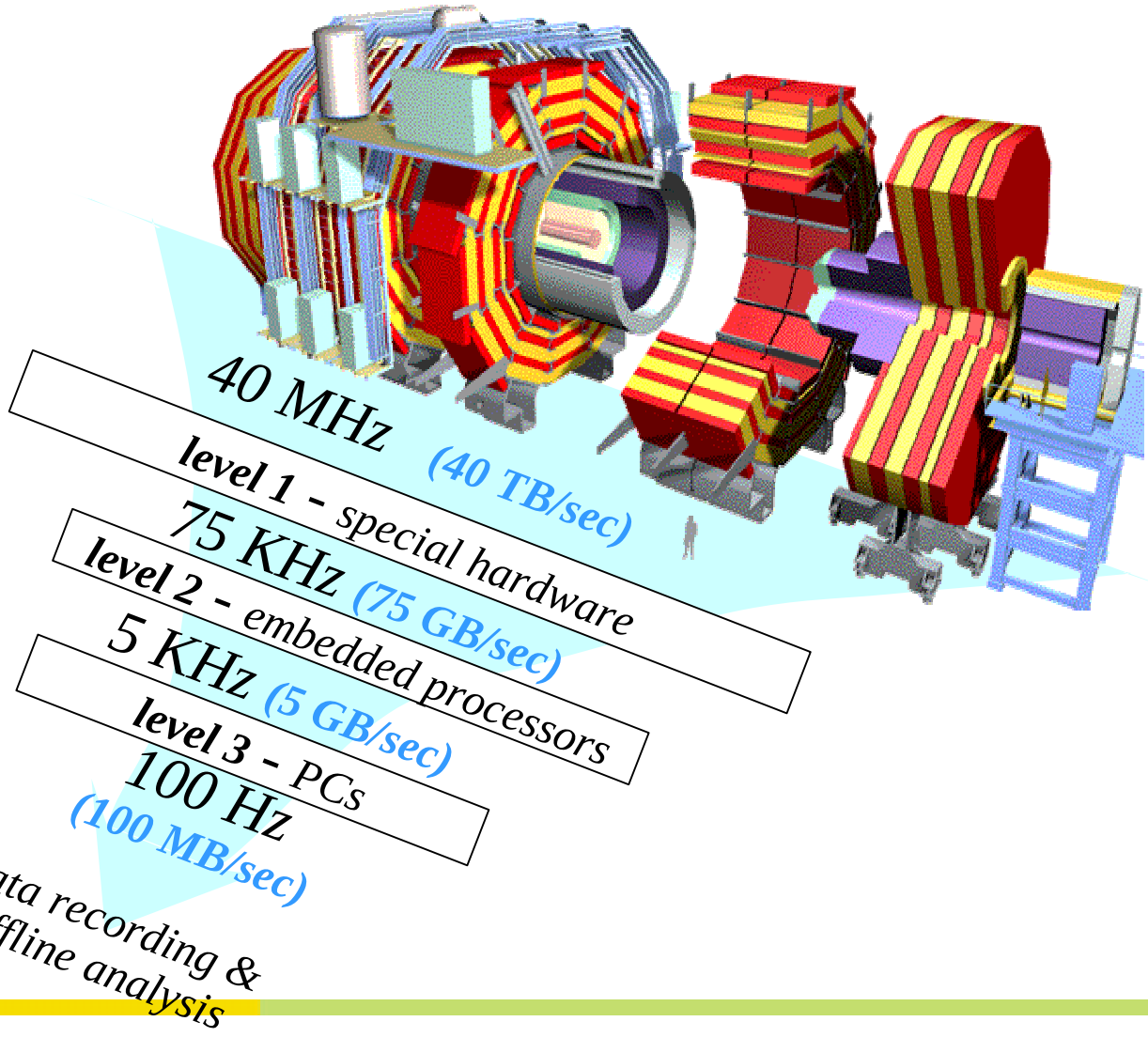
- To reconstruct and analyze 1 event takes about 90 seconds
- Most collisions don't result in observable "spark plug fragments" – could be as few as one out of a million. But we have to check them all!
- Computer program needs lots of "calibration"; determined from inspecting results of first pass.
  - Refine map of detector elements
  - Relation between detector signal strength and particle energy deposition
  - Calibrate detector clocks (how many ticks per microsecond?)
- Each event will be analyzed several times!

Data Handling and Computation for Physics Analysis



CMS
Compact Muon Solenoid

One of the four LHC detectors



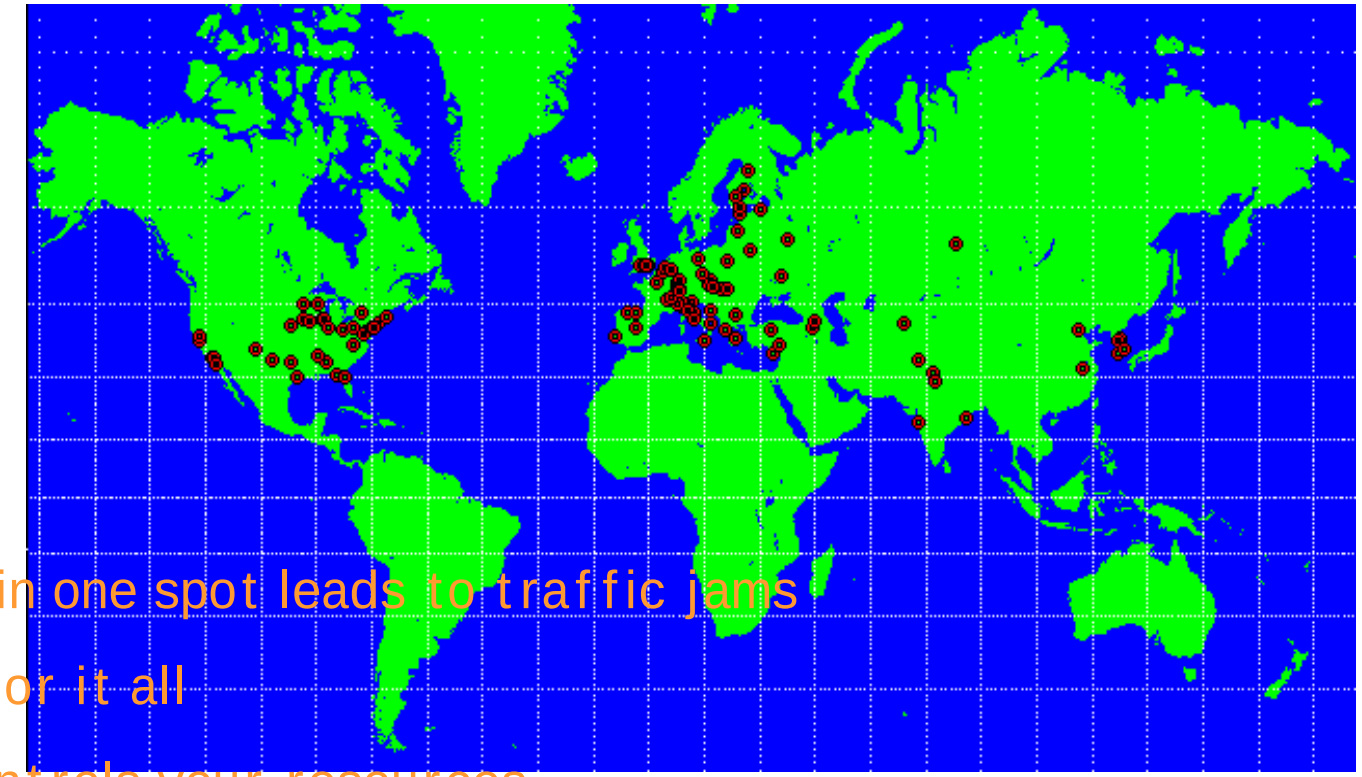
online system
multi-level trigger
filter out background
reduce data volume

Computational Implications (2)

- 90 seconds per event to reconstruct and analyze
- 100 incoming events per second
- To keep up, need either:
  - A computer that is *nine thousand times faster*, or
  - *nine thousand computers* working together
- Moore's Law: wait 21 years and computers will be 9000 times faster (we need them in 2006!)
- Grids: make large numbers of computers work together
- Four LHC experiments plus extra work: need >50k computers

A bunch of computers is not a Grid

- HEP has experience with a couple thousand computers in one place



BUT

- Putting them all in one spot leads to traffic jams
- CERN can't pay for it all
- Someone else controls your resources
- Can you use them for other (non-CERN) work?

Distribute computers like users

- Most of computer power not at CERN
  - need to move users' jobs to available CPU
  - data need to be "close" to CPU using them
- Need computing resource management
  - How to connect users with available power?
- Need data storage management
  - How to distribute?
  - What about copies? (Lots of people want access to same data)
- Need authorization & authentication for access to resources!

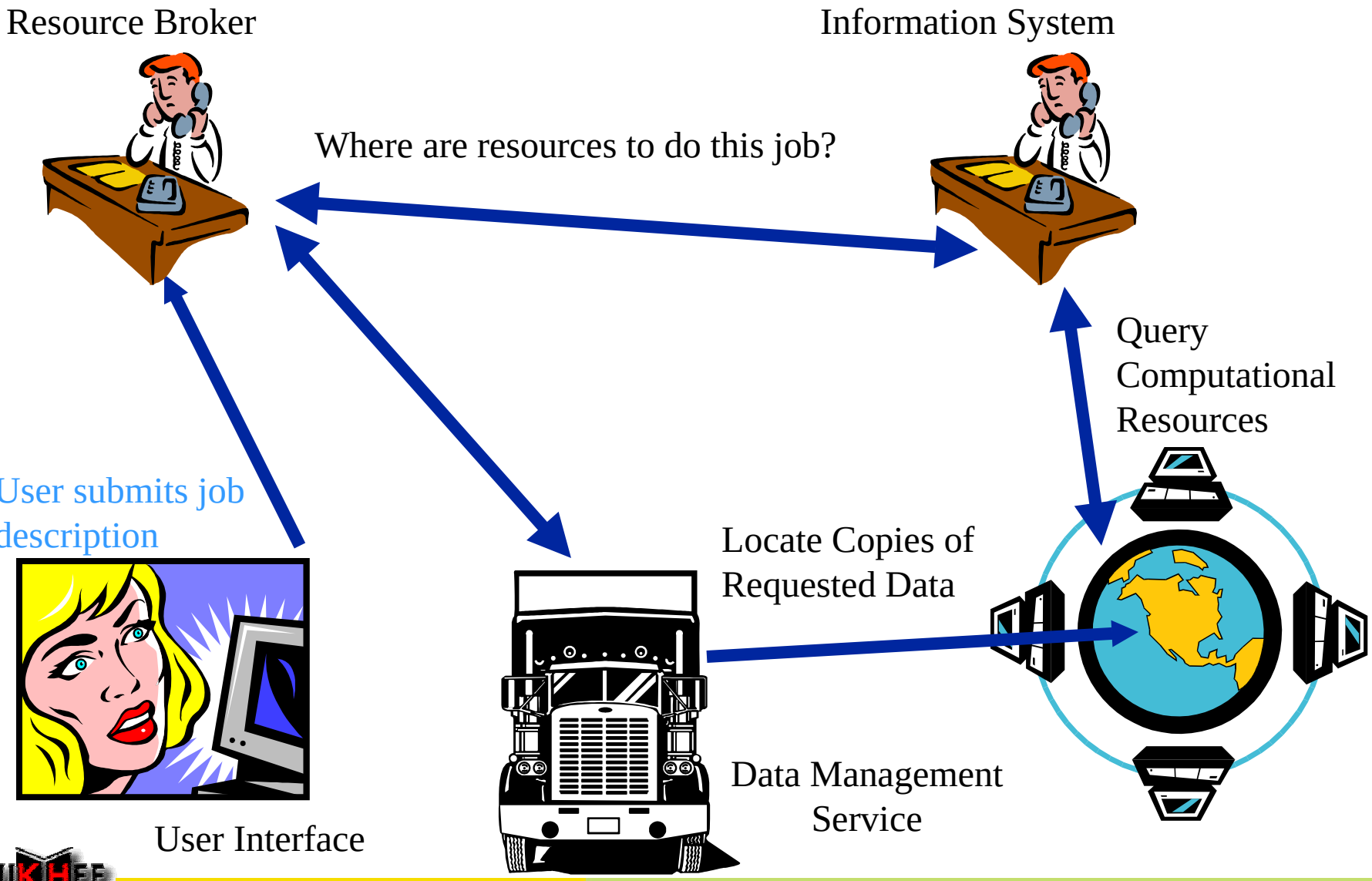
Grids: wide-area computing

- Grids implement distributed task scheduling and execution
- Grids implement distributed data
  - Storage
  - Access
  - Replication
  - Management
- Grids facilitate authentication, authorization, and accounting across national (continental, institutional) boundaries
- Grids give you potential access to 1000's of computers, but institutes can set their own priorities for their contribution: institutes "own" some of the resources

What does the Grid do for you?

- You submit your work, and the Grid
  - Finds convenient places for it to be run
  - Organises efficient access to your data
    - Caching, migration, replication
  - Deals with authentication to the different sites that you will be using
  - Interfaces to local site resource allocation mechanisms, policies
  - Runs your jobs
  - Monitors progress and recovers from problems
  - Tells you when your work is complete
- If your task allows, Grid can also decompose your work into convenient execution units based on available resources, data distribution

Grid Session: Matchmaking



Grid Session: Job Placement

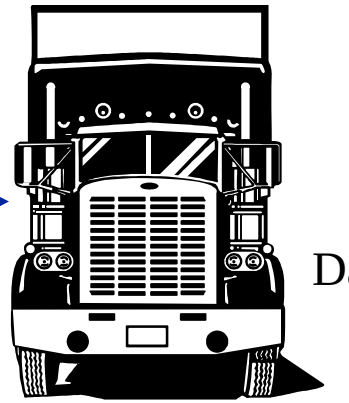
Resource Broker

Submit Job

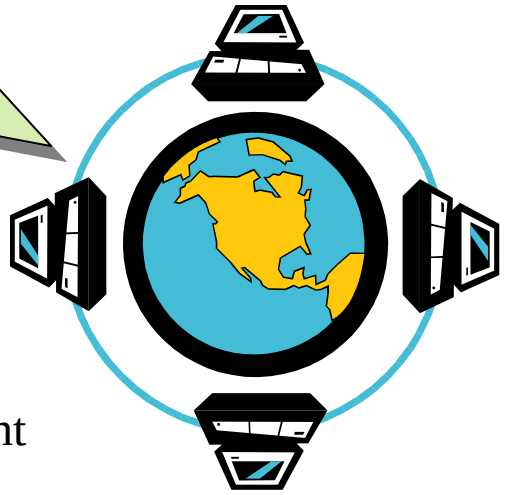
Resource Broker decides "optimal" place to run job

RAL

Resource Broker request movement of data to chosen site

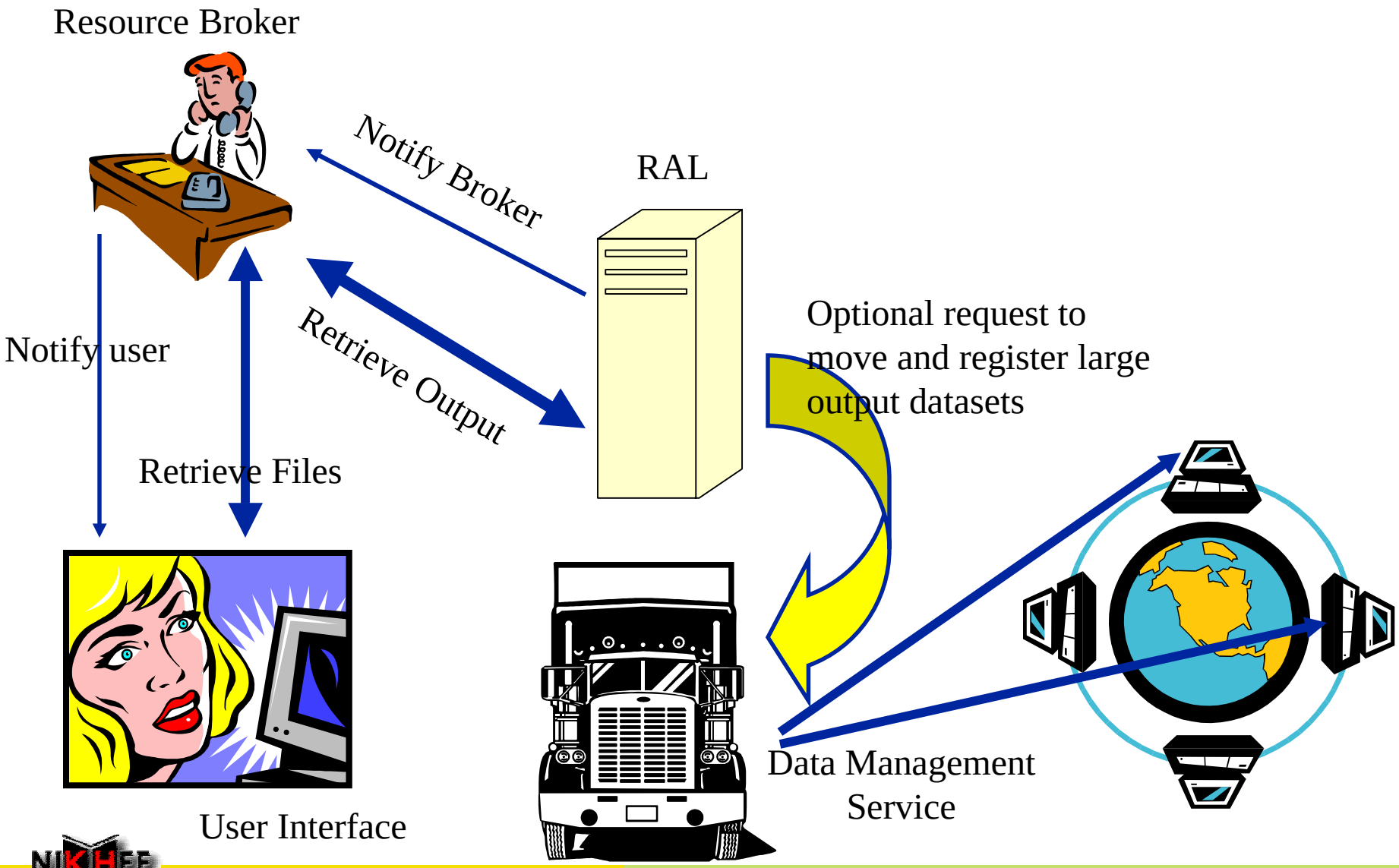


Data Management Service



- Available processors
- On-site (or "close") copies of job data
- User (or his "virtual organization) allowed to run there

Grid Session: Job Termination



What's There Now?

- Job Submission
  - Marriage of Globus and Condor-G – works relatively well
- Information System
  - Globus MDS (Metacomputing Directory Service)
    Problems with stability … planned to be replaced with R-GMA (product from DataGrid project)
- File Transfer
  - GridFTP – works very well and uses multiple internet connections to transfer files very quickly – can utilize up to 90% of available connection bandwidth
- Data Management
  - GDMP – very basic prototype, fragile, to be replaced shortly

More Stuff There Now

- Cluster Management
  - LCFG – extremely useful tool.
  - Used to manage about 25 machines at NIKHEF.
  - One server machine contains configuration for each machine type plus map of which machines should be what type
  - Each machine controlled by LCFG polls server every two minutes for new configuration information or software upgrades
  - Possible to reconfigure cluster completely in about 15 minutes (power fail story)
  - New machine? Little work, very quick

More Stuff There Now

- Networking
  - Bandwidth monitoring services nearly finished
    - Find out how "close" a computing center is to the data needed by a job
  - Lots of interesting monitoring tools
- Security
  - GSI from Globus – works quite well in practice
  - User obtains certificate from Nat'l Authority
    - "I am Jeff Templon"
    - Protected by passphrase
  - Certificate "subjects" distributed to places where JT has access
  - You can use your cert (from anywhere!) to access Grid services

More Stuff There Now

- Virtual Organizations
  - We have ten:
    - Four LHC experiments, two US HEP experiments
    - Bioinformatics
    - Earth Observation
    - Two for development activities
- Each site can
  - Decide whether to accept individual VOs
  - Assign priorities to VOs
- Some services have copies for each VO (e.g. Data Management)

What is on the horizon

- True Replica Management
  - Distributed Replica Catalog – each grid site keeps list of datasets present locally, with fast transparent access to lists from other sites
  - Data Management at job submission – Resource Broker commands Data Management Service to move files to support user jobs
  - Strategic Data Management – Service keeps track of "who accessed what data from where" and makes automatic movement to improve job performance
- Mass Storage Support
  - Make mass storage (e.g. tape robots) invisible for user

What we're missing

- How to do automatic program decomposition
  - HEP has big files full of events
  - Would like Grid to break up job into several pieces – as many pieces as there are available processors!
- Grid needs to know something about how to decompose
  - Your file is just a bunch of bits unless you tell the Grid how to read it
- Similar problems for "true parallel" jobs
  - How to distribute on-the-fly based on number of nodes available?
  - Are there efficient high-latency algorithms out there?

What's been hard

- Collaboration – distributed software construction is hard
- Make services work together without making them codependent
- "Paratrooper Programming" … current software survives only in controlled environment

Trends to Watch

- Opportunistic Scheduling
  - Condor project – install Grid software on desktop PCs, let outside users take spare cycles. We have 171 desktop Linux systems at NIKHEF, and mine was 98.6% idle when I wrote this
- Web Services
  - Current Grid services are accessed over internet and advertised in information system; programs using service must already "know how" to do it
  - Web services: service registers with an information system (service registry)
  - Tells registry "this is how a program is supposed to use my service"
  - Sent as XML description to client programs

Example: File Transfer

- Suppose my program needs to transfer output to some other machine (server)
- Current situation: the worker node (where my program runs) needs to be preprogrammed for all expected protocols on all servers on all machines
- Web Services: the worker-node file transfer program must be able to understand XML
- Service registry provides
  - List of data transfer services provided by target machine
  - instructions (via XML) on how to use protocol each service implements
- Client program contacts selected service per prescription
- Grid version called “OGSA”, collaboration between Globus project and IBM (with support from NASA Information Power Grid)

Conclusions

- Grids well-suited to providing HEP computing power
- Grids have advantages for strategic sharing of local and remote computing resources
- We have quite a bit working already (European DataGrid project)
- Still learning how to make “paratrooper programs”
- Will be very interesting to see if Web Service concept lives up to expectations