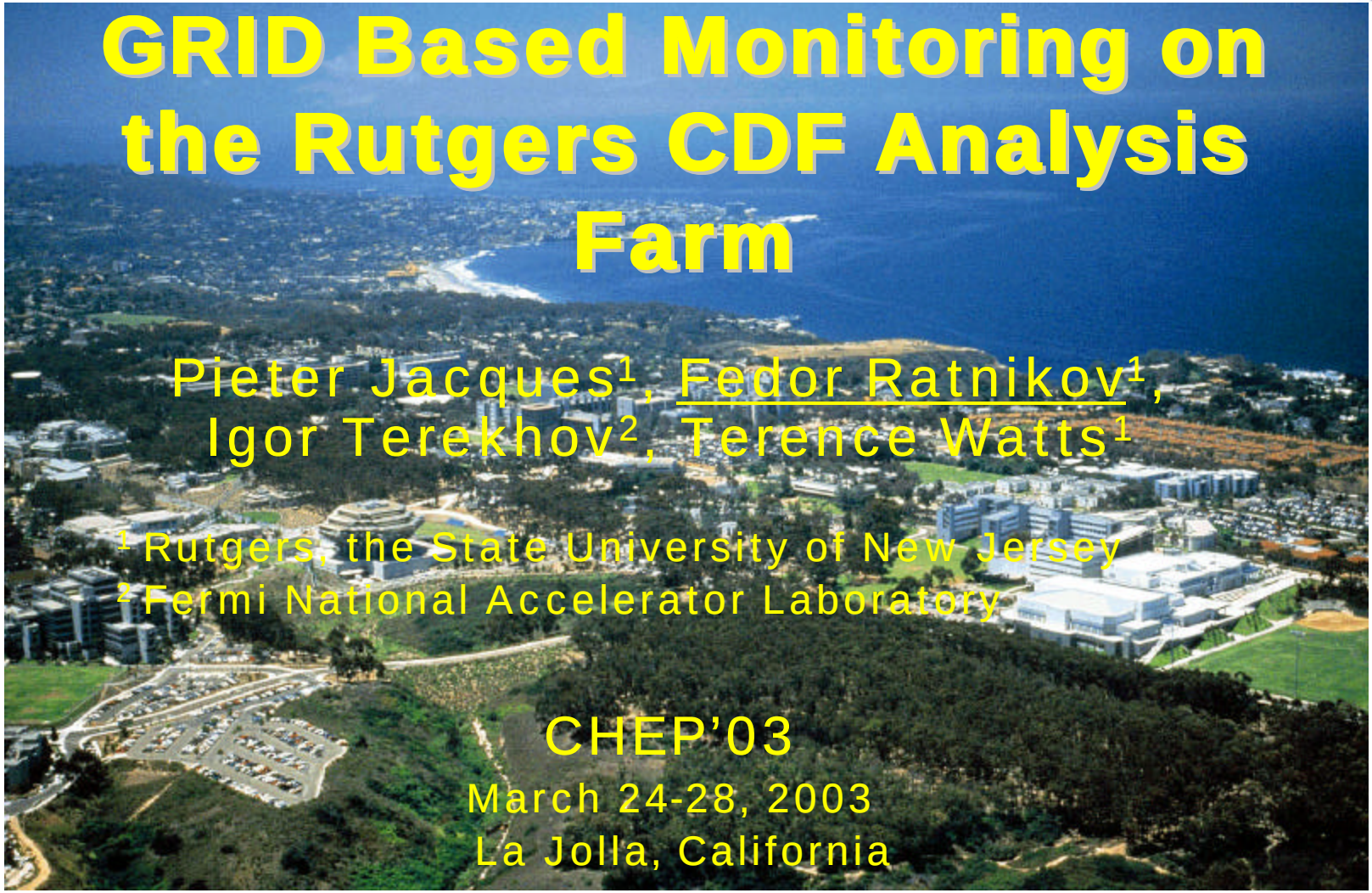


GRID Based Monitoring on the Rutgers CDF Analysis Farm

Pieter Jacques[1], Fedor Ratnikov[1], Igor Terekhov[2], Terence Watts[1]

[1] Rutgers, the State University of New Jersey
[2] Fermi National Accelerator Laboratory

CHEP'03
March 24-28, 2003
La Jolla, California

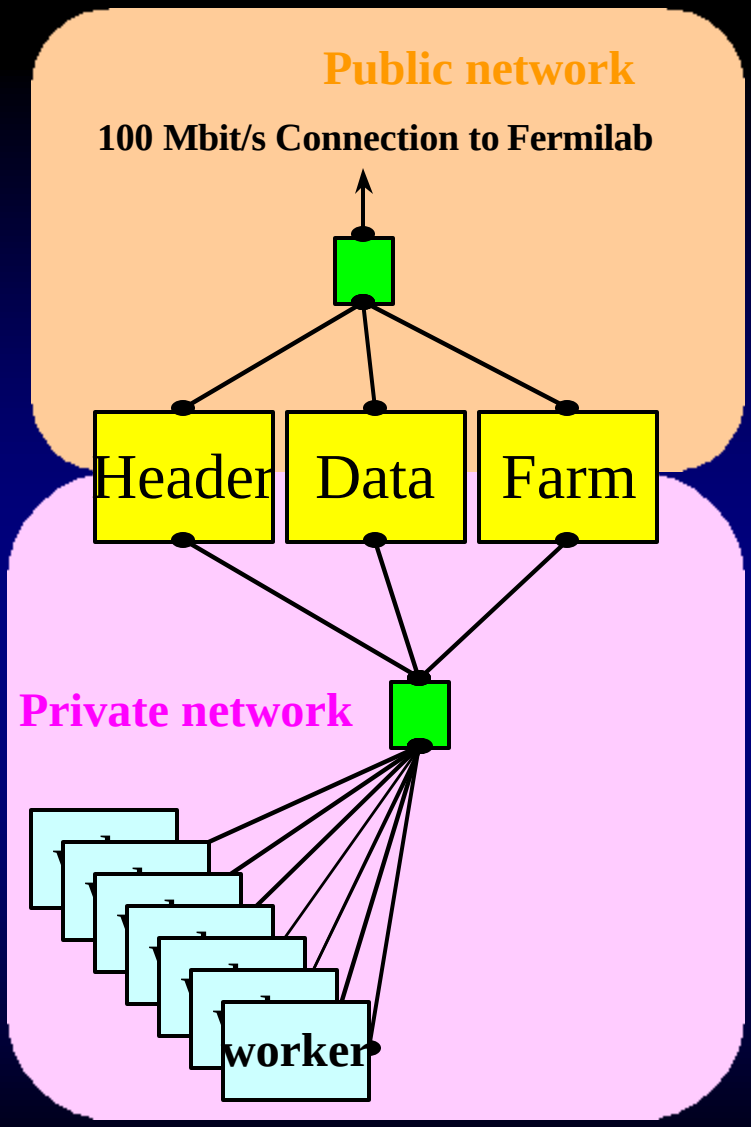
HEX Farm @ Rutgers University

- HEP group ~20 physicists participating in
  - 70% CDF
  - 15% CMS
  - 15% others
- Primary goal of the HEX farm is to satisfy computing needs of the CDF group for physics data analysis

➔ This talk will show operational issues of running GRID-like environment on the university analysis farm (not production farm)

HEX Farm Hardware

- Dual CPU PCs running Fermi Linux
- Header node
  - Interactive node
  - Hosts home directories and experiment specific software
  - NAT server
  - NIS server
- Data transfer node
  - Pass data traffic between public and private network
- Farm header node
  - Controls farm load
  - Runs SAM station
  - Hosts SAM cache
  - The only fully kerberized node in the system
- ~15 worker nodes on private network
- Big IDE data disks connected to worker nodes are cross mounted on every node of the farm



Resource Management

- Use CDF approved components
  - Batch system
    - CDF developed CAF(*) built on top of Fermilab's FBSNG
  - Data Handling
    - Disk cache only, no local mass storage
      - Rely on mass data storage @Fermilab
    - Several different data delivery and cataloguing systems coexist on the farm
      - ➔CDF native DIM
      - ➔D0 native SAM
      - ➔Direct remote access dCache

(*) See details in C-3 presentation of Frank Wuersthwein

Data Handling Systems Variety (I)

- Light weighted CDF baseline Disk Inventory Manager (DIM) (*)
  - Natively supported by CDF software
  - Import data from central DIM @ Fermilab
  - Independent MSQL Data File Catalog
  - Used both as a storage for static datasets and as a dynamic data cache
    - Automatic synchronization of static datasets with primary copies @ Fermilab when new data are added to the dataset
    - Automatic delivery of dynamic data into data cache
  - Flexible transport protocol (dCache + GridFtp currently)
  - Data transfer unit is a fileset (~10 files $\times$ 1GB)

(*) See details in C-8 presentation of Dmitry Litvinsev

Data Handling Systems Variety (II)

- Sequential data Access via Meta-data (SAM) – sophisticated baseline DH system for D0. CDF is currently in a process of accommodating SAM (*)
  - Local data cache
  - Central data catalog
  - Automated data delivery – transfer unit is a file
  - Supports parallel processing of a data sample
    - SAM is supposed to control the job submission
  - Provides native mechanisms for storing and cataloguing output data in the central mass storage system

(*) See details in C-1 presentations of Gabriele Garzoglio and Lee Lueking and C-2 presentation of Stefan Stonjek

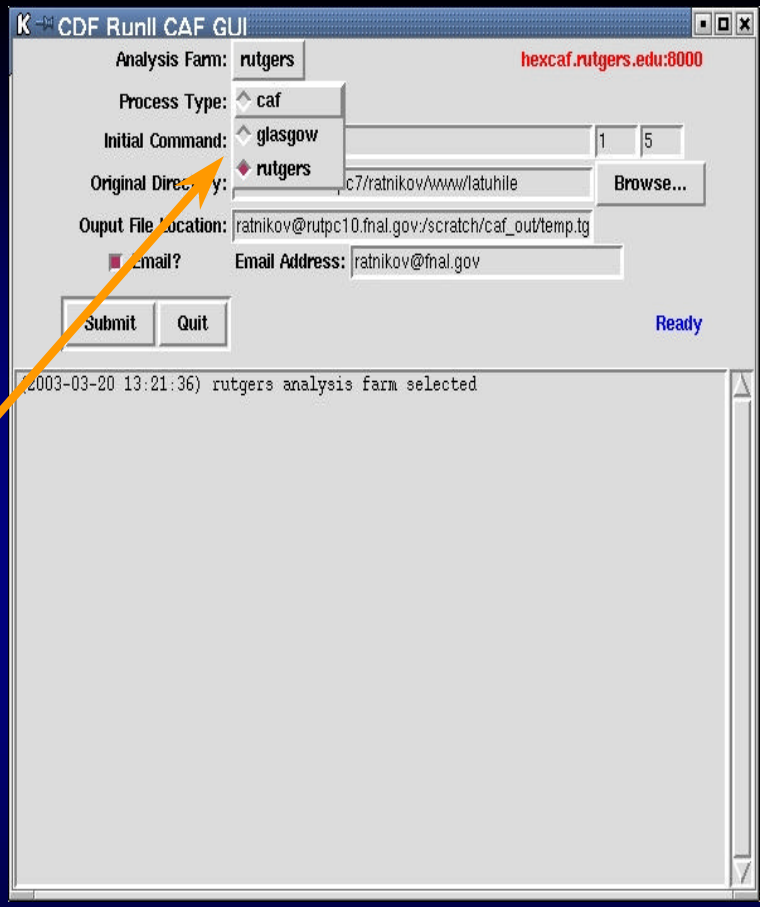
Data Handling Systems Variety (III)

- Direct access of central data @Fermilab via dCache (*)
  - Natively supported by CDF software
  - Remote data cache @ Fermilab
  - Files are opened remotely
    - Only required part of the information is transmitted via network
  - Simple in use
    - Bypasses any local DH system

(*) See details in C-8 presentation of Robert Kennedy

GRID Like Approach

- CAF philosophy fits well the analysis farm requirements
  - Assumptions
    - Baseline software is available on the farm
    - Job tarball is made on the fly and contains everything to run in the baseline environment
    - Output is securely delivered back to user
  - Submission, execution and output delivery are well decoupled
    - Universal submitter allows easy redirection of jobs to any farm
    - Naturally extendable for the resource broker driven operations



True GRID Approach

- GRID Job and Information Manager (JIM, aka SAMGRID) (*)
  - Resource broking
  - Job submission
  - Monitoring
  - Built on top of Condor-G
  - Uses SAM information about data files availability to select a destination for the job
  - Has an adapter to the local CAF submitter
  - LDAP based information providers and monitoring tools

(*) See details in C-1 presentations of Igor Terekhov and Gabriele Garzoglio

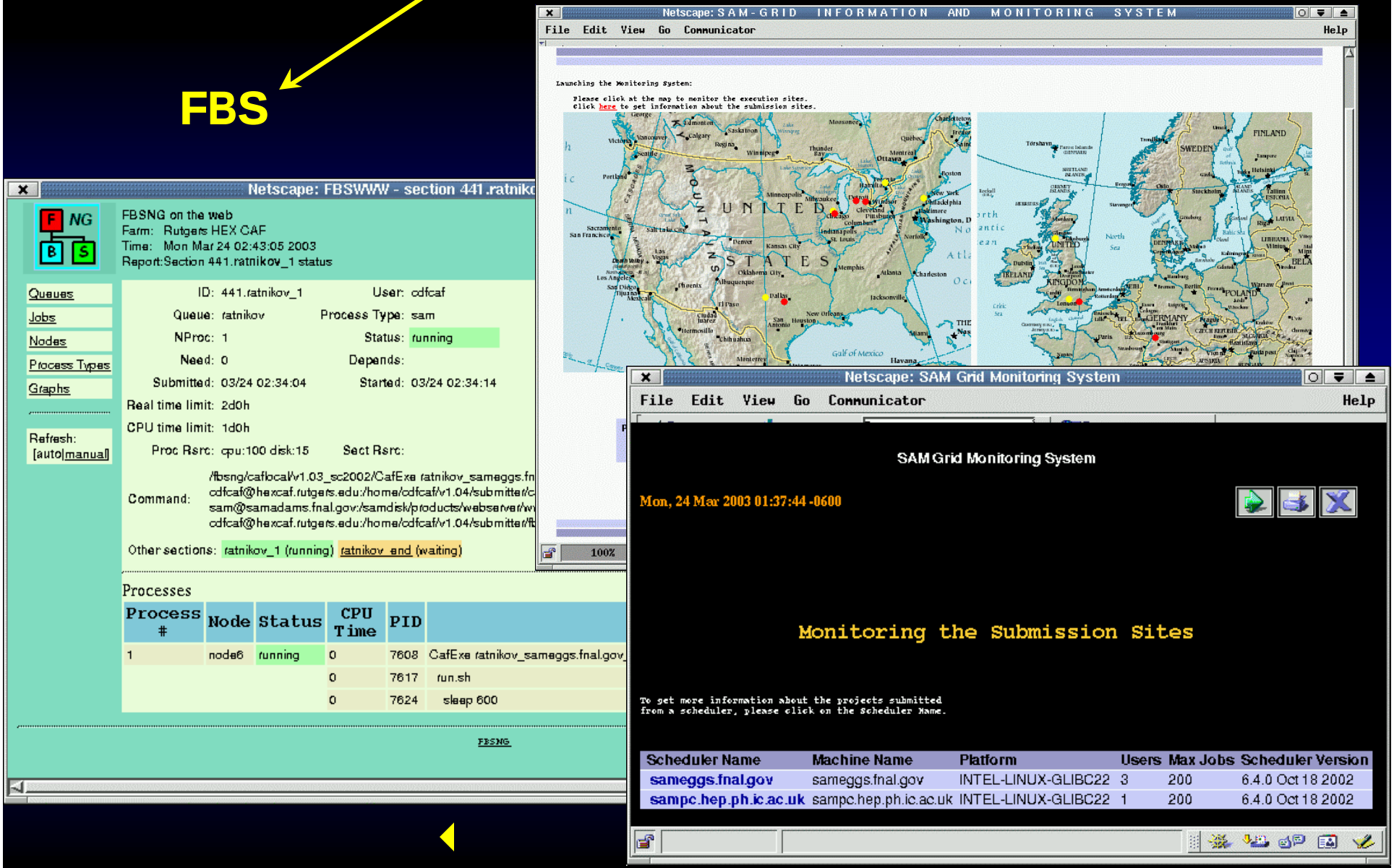
Monitoring

- Both JIM and CAF (FBSNG) have nice monitoring systems available
  - But they are not aware about each other
  - Need a bridge from the global (Grid-like) system to the local batch system
- Database driven solution is selected for JIM and implemented on the HEX farm

Monitoring

JIM

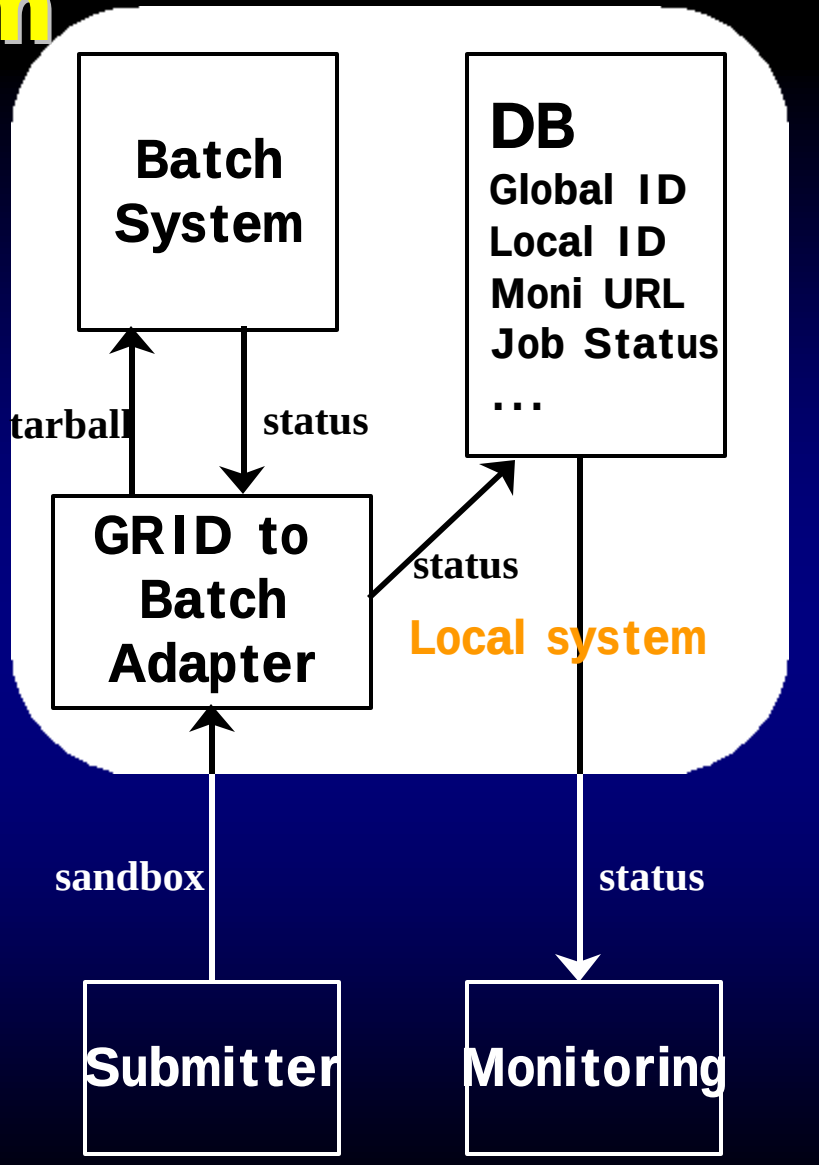
FBS



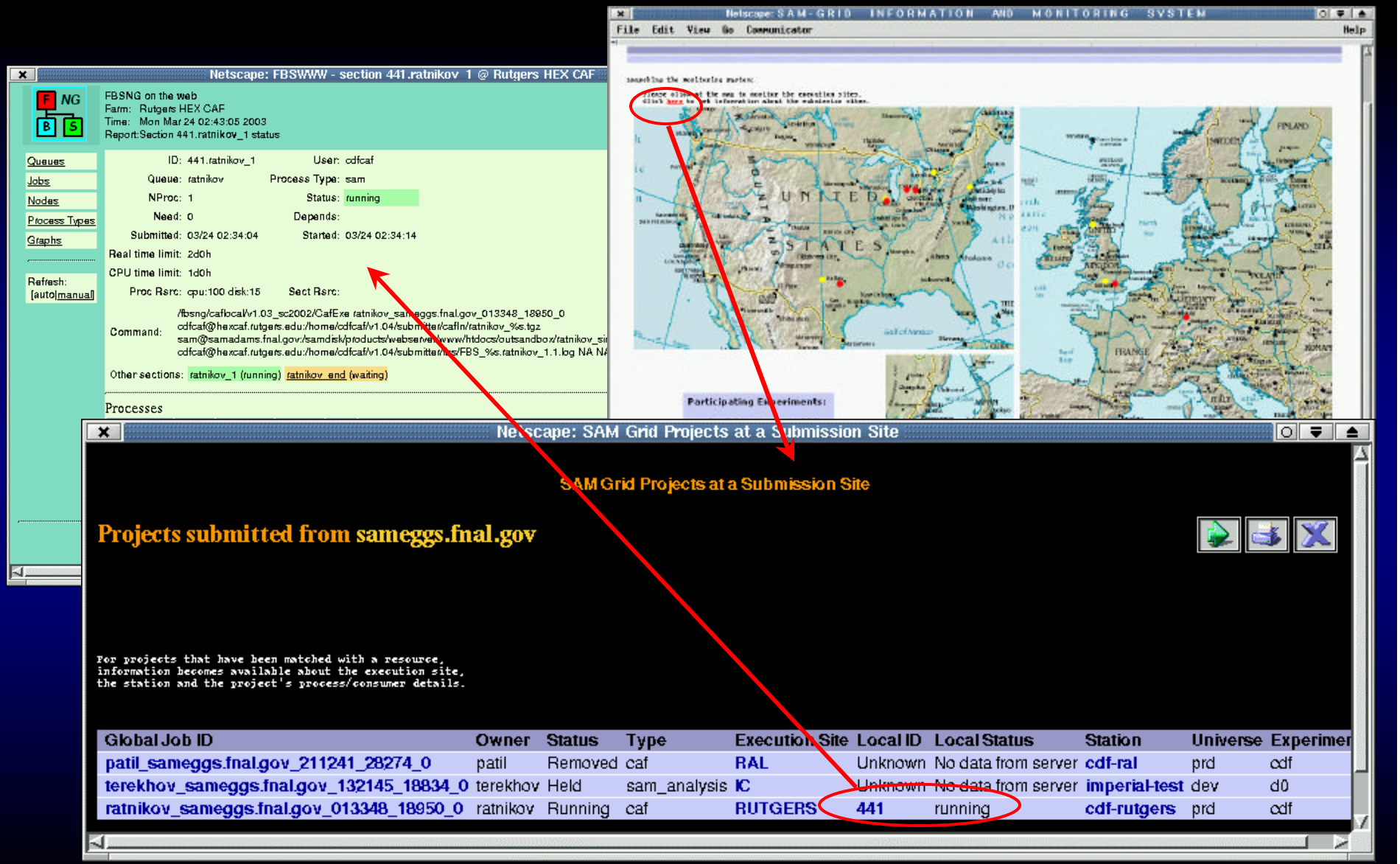
Global Monitoring of the Local System

- Grid-to-local batch adapter fills the local database with a global job ID and corresponding local batch job references
- Local Information Provider extracts information from the DB and reports it to the outside client

- Benefits
  - Decoupled batch system
  - Single global-to-local adaptor
  - Uniform information access



Combined Monitoring → JIM → FBS



Setup Summary

- HEX serves a typical university group actively participating in big HEP experiments
- A variety of different data access and job submission systems are installed and coexist
- Convenient monitoring is provided for operating components
- Users are free to select any approach for their work

- What do users choose?

Operation Experience

- 80% of data are ~10 static DIM datasets resident on the disk. These data are intensively used for data analysis
- 20% of data are accessed via dynamic DIM driven cache
- 100% of jobs are submitted via standard CAF submitter

☝ The farm is used by the group members for the physics analysis:
  - No massive data production
  - No massive MC production
  - Static data can easily be managed manually in such small community

☝ DIM is in use by CDF for a long time, so it is the most familiar to people

✓ Users vote for a simple and straightforward solution that can give immediate outcome and help making physics results faster
  - Even perfect resources management is not appreciated if it requires too much efforts to be understood and used

Conclusions

- Rutgers HEX farm is kept tuned and uses most advanced computing techniques deployed for the CDF collaboration
- Farm is intensively used for testing and development of modern distributed computing technologies
- Users are not forced to use any particular approach, they can select the most convenient one from available variety
- Users will step on the new technology immediately as soon as this technology will save their time and speed up physics analysis
- ... and not earlier