DØ Monte Carlo Challenge

A HEP Application

Outline

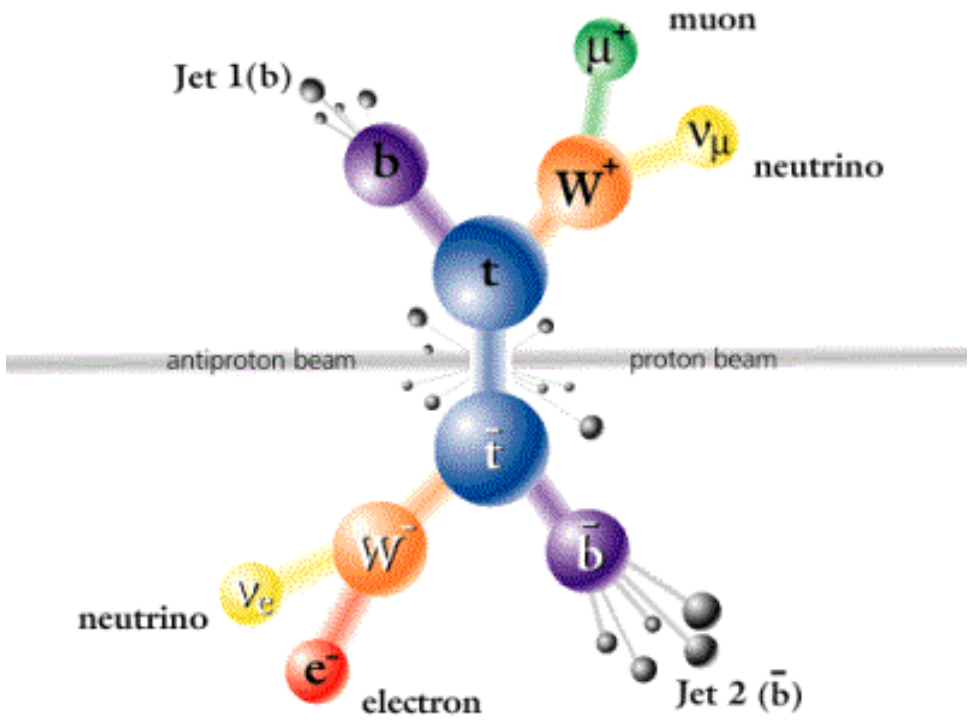
- The DØ experiment
- The application
- The NIKHEF DØ farm
- SAM (aka the DØ grid)
- Conclusions

The DØ experiment

- Fermi National Accelerator Lab
- Tevatron
  - Collides protons and antiprotons of 980 GeV/c
  - Run II
- DØ detector
- DØ collaboration
  - 500 physicists, 72 institutions, 19 countries

The DØ experiment

- Detector Data
  - 1,000,000 Channels
  - Event size 250KB
  - Event rate ~50 Hz
  - On-line Data Rate 12 MBps
  - Est. 2 year totals (incl Processing and analysis):
    - 1 x 10^9 events
    - ~0.5 PB
- Monte Carlo Data
  - 5 remote processing centers
  - Estimate ~300 TB in 2 years.


muon
Jet 1(b)
neutrino
antiproton beam
proton beam
neutrino
electron
Jet 2 ($\bar{b}$)

The application

- Generate events
- Follow particles through detector
- Simulate detector response
- Reconstruct tracks
- Analyse results

The application

- Starts with the specification of the events
- Generates (intermediate) data
- Stores data in tape robots
- Declares files in database

The application

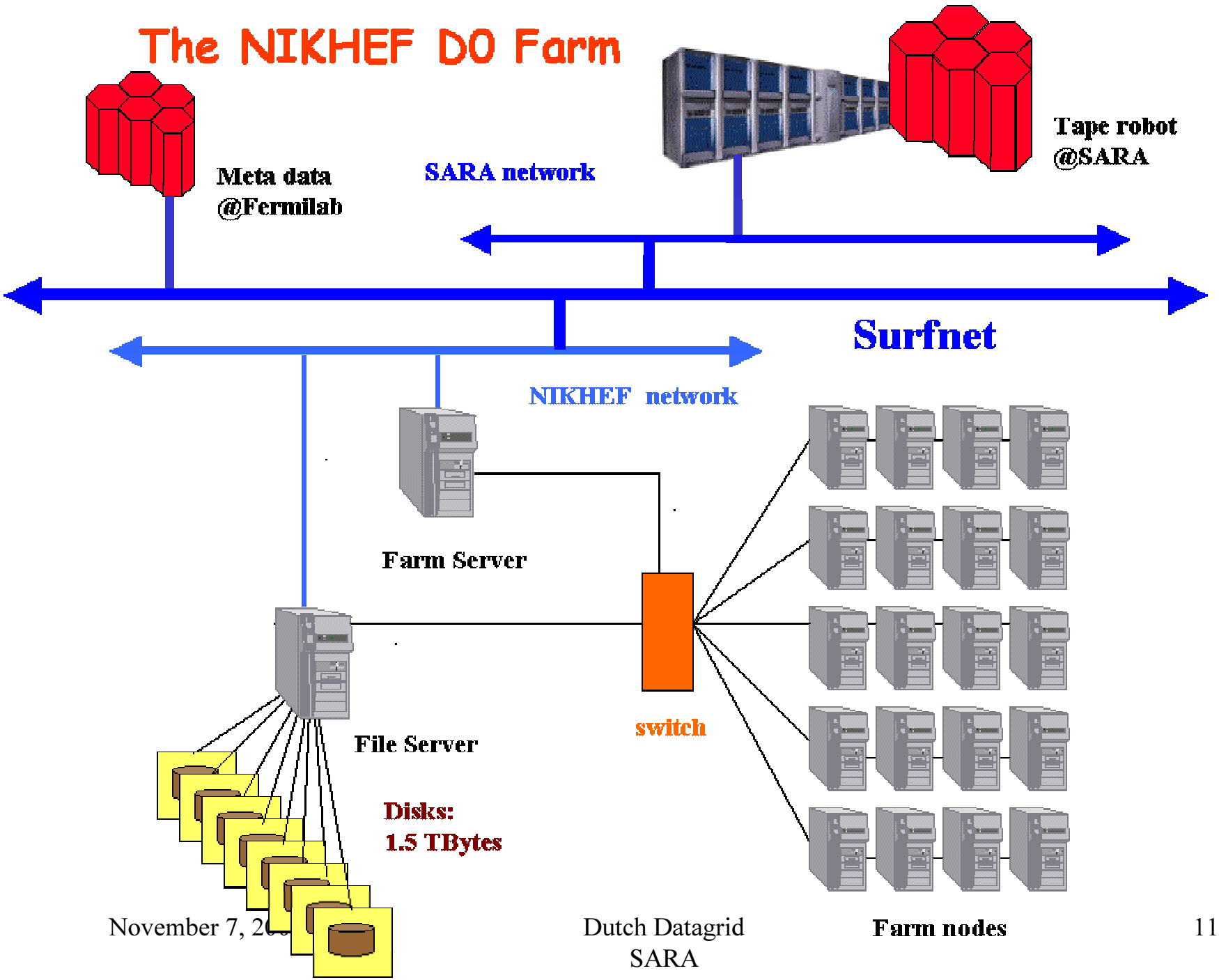
- consists of
  - Monte Carlo programs
    - gen, d0gstar, sim, reco, recoanalyze
  - mc_runjob
    - bunch of python scripts
- runs on
  - SGI Origin (Fermilab, SARA)
  - Linux farms

mc_runjob

- Creates directory structure for job
- Creates scripts for each jobstep
- Creates scripts for submission of metadata
- Creates job description file
- Submit job to batch system

The NIKHEF DØ farm

- Batch server (hoeve)
  - Boot/Software server
  - Runs mc_runjob
- File server (schuur)
  - Runs SAM
- 50 – 70 nodes
  - Run MC jobs

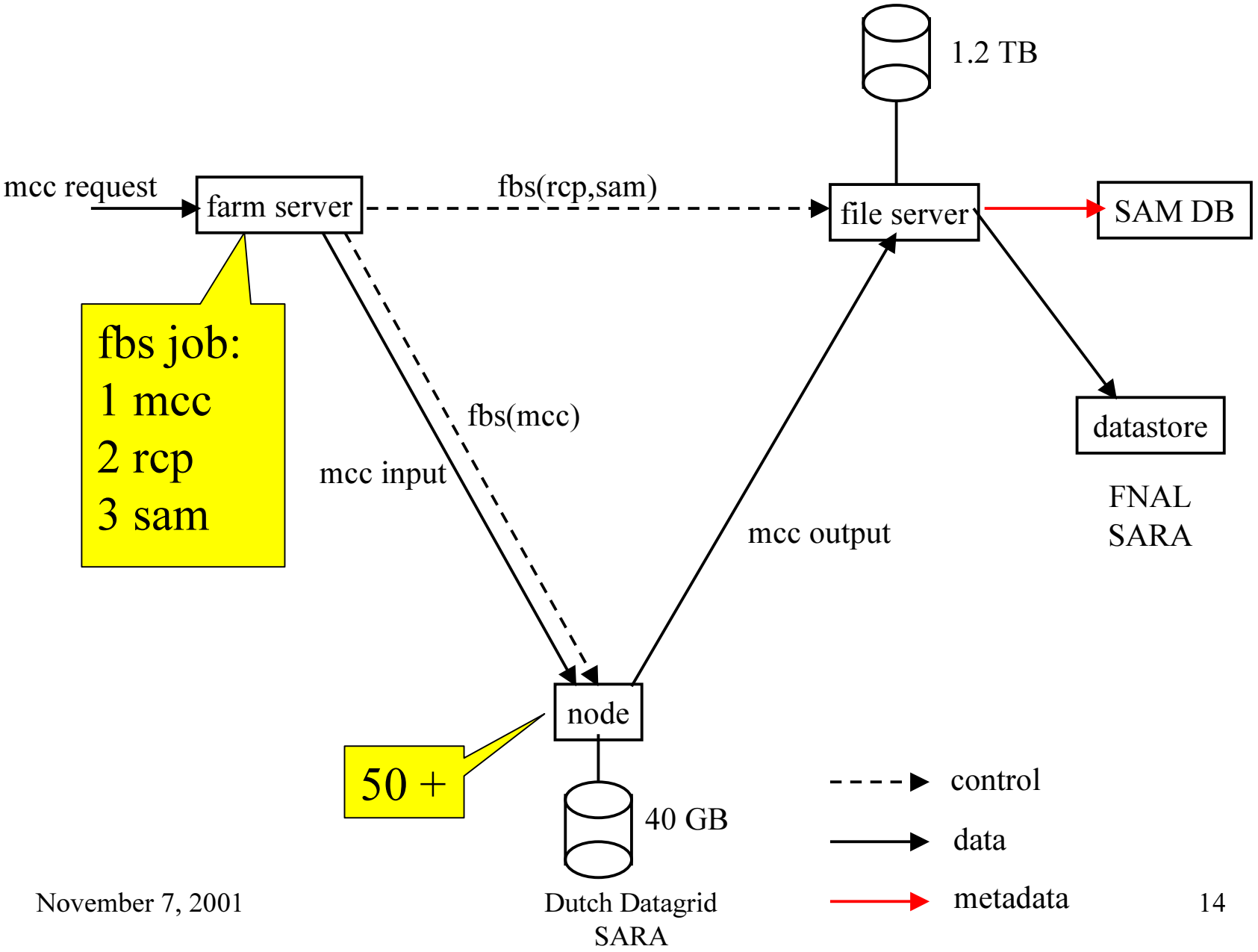

The NIKHEF D0 Farm
Meta data
@Fermilab
SARA network
Tape robot
@SARA
Surfnet
NIKHEF network
Farm Server
File Server
Disks:
1.5 TBytes
switch
Farm nodes

node

- At boottime:
  - Boots via network from batch server
  - NFS mounts DØ directories on batch server
- At runtime:
  - Copies input from batch server to local disk
  - Runs MC job steps
  - Stores (intermediate) output on local disk

File server

- Copies output from node to file server
- Declares files to SAM
- Stores files with SAM in robot
  - @ fnal
  - @ sara


1.2 TB
mcc request
farm server
fbs(rcp,sam)
file server
SAM DB
fbs job:
1 mcc
2 rcp
3 sam
fbs(mcc)
mcc input
mcc output
datastore
FNAL
SARA
node
50 +
40 GB
control
data
metadata
Dutch Datagrid
SARA

SAM @ NIKHEF

- Stores metadata in database at FNAL
  - sam declare import_<jobstep>.py
  - scripts prepared by mc_runjob
- Stores files
  - on tape at fnal via cache on d0mino
  - on disk of teras.sara.nl and migrated to tape
  - sam store --descrip=import_<jobstep>.py [--dest=teras.sara.nl:/sam/samdata/y01/w42]

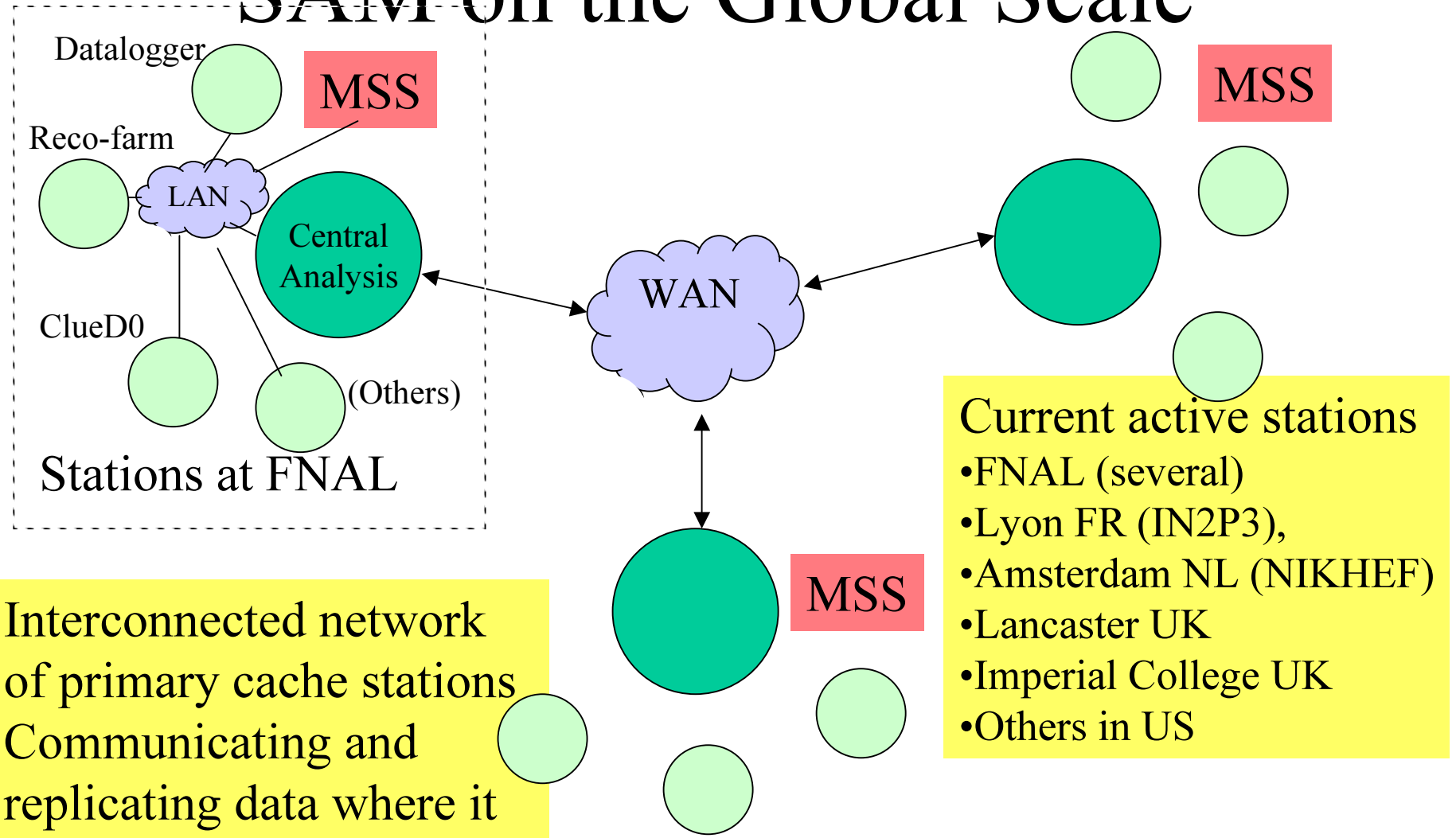
SAM @ SARA

- No need to install SAM ☺
- Declare teras directories in SAM as destination
- Access protocol
  - May 2001 rcp
  - October 2001 bbftp
  - ??: gridftp

SAM on the Global Scale

- Locate files
  - Monte Carlo data
  - Raw data from detector
  - Calibration data
  - Accelerator data
- Submit (analysis) jobs on local station
- Stores results in SAM

SAM on the Global Scale



Interconnected network of primary cache stations Communicating and replicating data where it is needed.

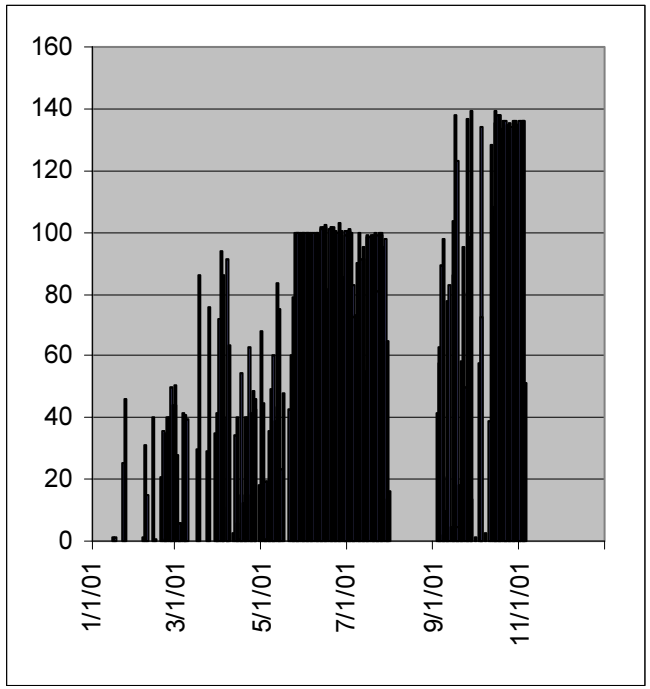
Current active stations
- FNAL (several)
- Lyon FR (IN2P3),
- Amsterdam NL (NIKHEF)
- Lancaster UK
- Imperial College UK
- Others in US

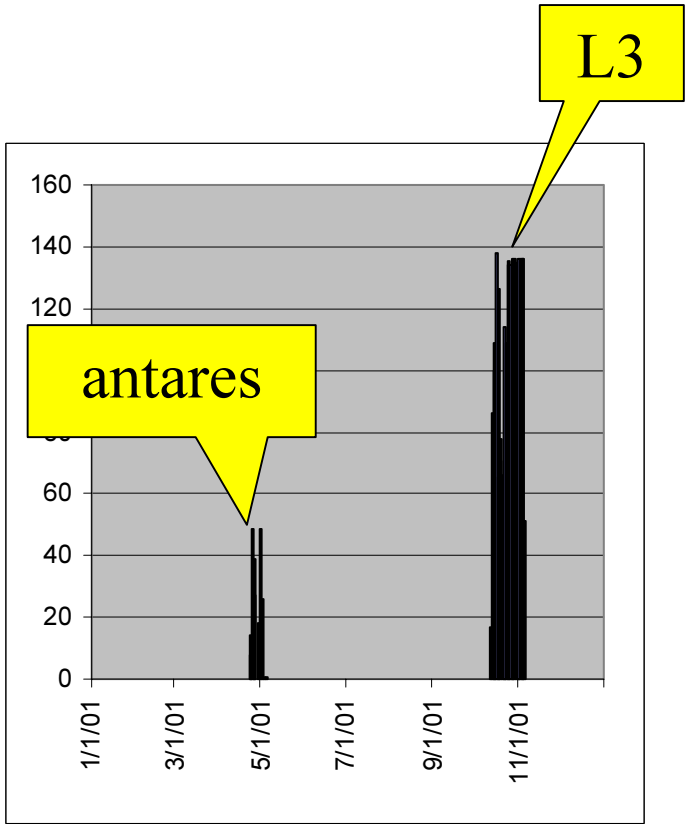
Future Plans for SAM

- Better specification of remote data storage locations, especially in MSS.
- Universal user registration that allows different usernames, uid, etc. on various stations.
- Integration with additional analysis frameworks, Root in particular (almost ready).
- Event level access to data.
- Movement toward Grid components, GridFTP, GSI…

Conclusions

- NIKHEF DØ farm is
  - easy to use (Antares, L3)
  - easy to clone (KUN)
  - part of DØ data grid
  - moving (slowly) to grid standards


160
140
120
100
80
60
40
20
0
1/1/01
3/1/01
5/1/01
7/1/01
9/1/01
11/1/01


L3
antares
160
140
120
60
40
20
0
1/1/01
3/1/01
5/1/01
7/1/01
9/1/01
11/1/01