



# High Throughput Distributed Computing - 1

**Stephen Wolbers, Fermilab**  
**Heidi Schellman, Northwestern U.**



# Outline - Lecture 1

- **Overview, Analyzing the Problem**
- **Categories of Problems to analyze**
  - "Level 3" Software Trigger Decisions
  - Event Simulation
  - Data Reconstruction
  - Splitting/reorganizing datasets
  - Analysis of final datasets
- **Examples of large offline systems**



## What is the Goal?

- Physics - the understanding of the nature of matter and energy.
  - How do we go about achieving that?
    - Big accelerators, high energy collisions
    - Huge detectors, very sophisticated
    - Massive amounts of data
    - Computing to figure it all out
- } These Lectures



*In Silica Fertilization*

# All Science Is Computer Science

By GEORGE JOHNSON

**E**XCEPT for the fact that everything, including DNA and proteins, is made from quarks, particle physics and biology don't seem to have a lot in common. One science uses mammoth particle accelerators to explore the subatomic world; the other uses petri dishes, centrifuges and other laboratory paraphernalia to study the chemistry of life. But there is one tool both have come to find indispensable: supercomputers powerful enough to sift through piles of data that would crush the unaided mind.

Last month both physicists and biologists made announcements that challenged the tenets of their fields. Though different in every other way, both discoveries relied on the kind of intense computer power that would have been impossible to marshal just a few years ago. In fact, as research on so many fronts is becoming increasingly dependent on computation, all science, it seems, is becoming computer science.

"Physics is almost entirely computational now," said Thomas B. Kepler, vice president for academic affairs at the Santa Fe Institute, a multidisciplinary research center in New Mexico. "Nobody would dream of doing these big accelerator experiments without a tremendous amount of computer power to analyze the data."

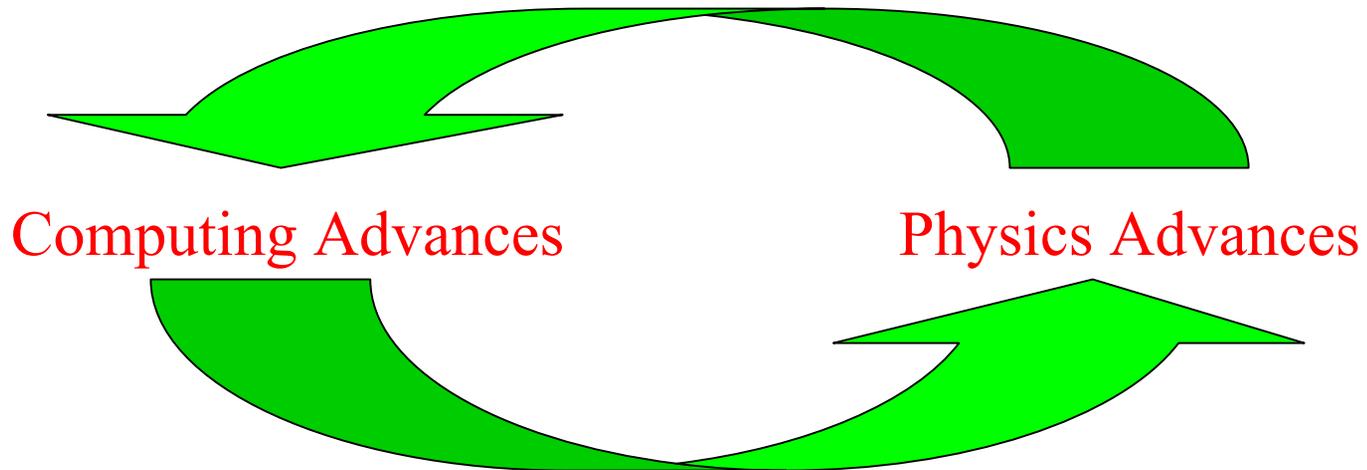
New York Times,  
Sunday, March 25, 2001

September, 2001



# Computing and Particle Physics Advances

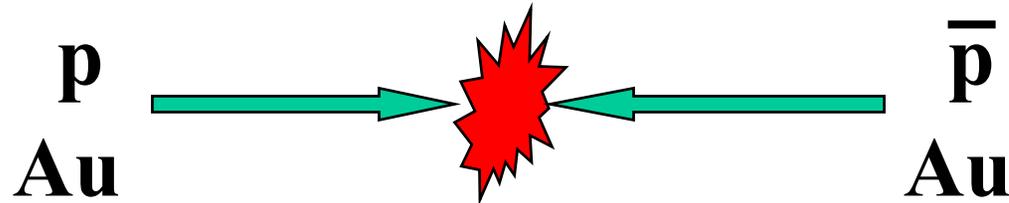
- HEP has always required substantial computing resources
  - Computing advances have enabled "better physics"
  - Physics research demands further computing advances
  - Physics and computing have worked together over the years



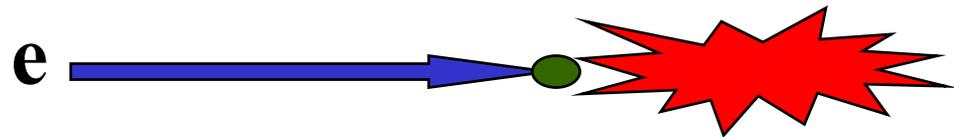


## Collisions Simplified

- Collider:

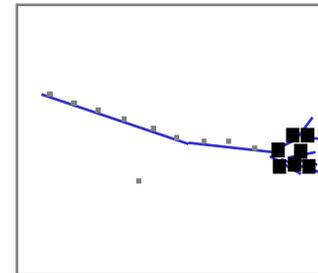
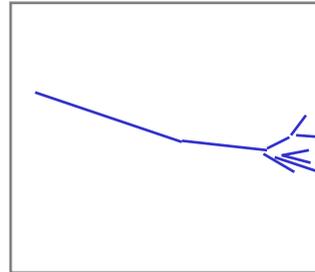
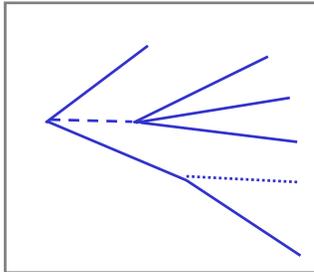
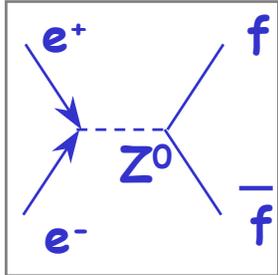


- Fixed-Target:





# Physics to Raw Data (taken from Hans Hoffman, CERN)



```
2037 2446 1733 1699
4003 3611 952 1328
2132 1870 2093 3271
4732 1102 2491 3216
2421 1211 2319 2133
3451 1942 1121 3429
3742 1288 2343 7142
```

Fragmentation,  
Decay

Interaction with  
detector material  
Multiple scattering,  
interactions

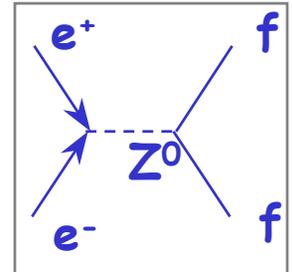
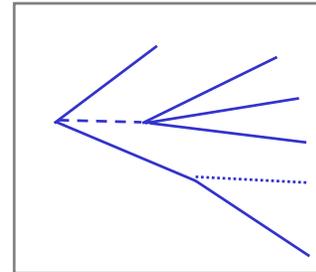
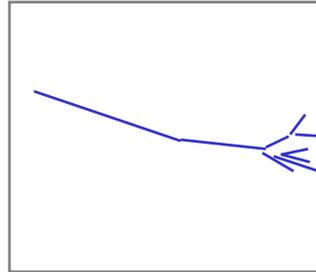
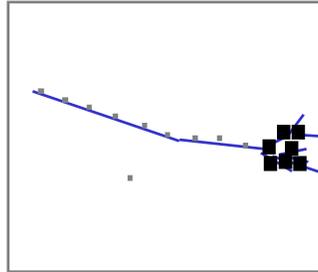
Detector  
response  
Noise, pile-up,  
cross-talk,  
inefficiency,  
ambiguity,  
resolution,  
response  
function,  
alignment,  
temperature

Raw data  
(Bytes)  
Read-out  
addresses,  
ADC, TDC  
values,  
Bit patterns



# From Raw Data to Physics

2037 2446 1733 1699  
4003 3611 952 1328  
2132 1870 2093 3271  
4732 1102 2491 3216  
2421 1211 2319 2133  
3451 1942 1121 3429  
3742 1288 2343 7142



Raw data

Convert to  
physics  
quantities

Detector  
response  
apply  
calibration,  
alignment,

Interaction with  
detector material  
Pattern,  
recognition,  
Particle  
identification

Fragmentation, Basic physics  
Decay  
Physics  
analysis

Results

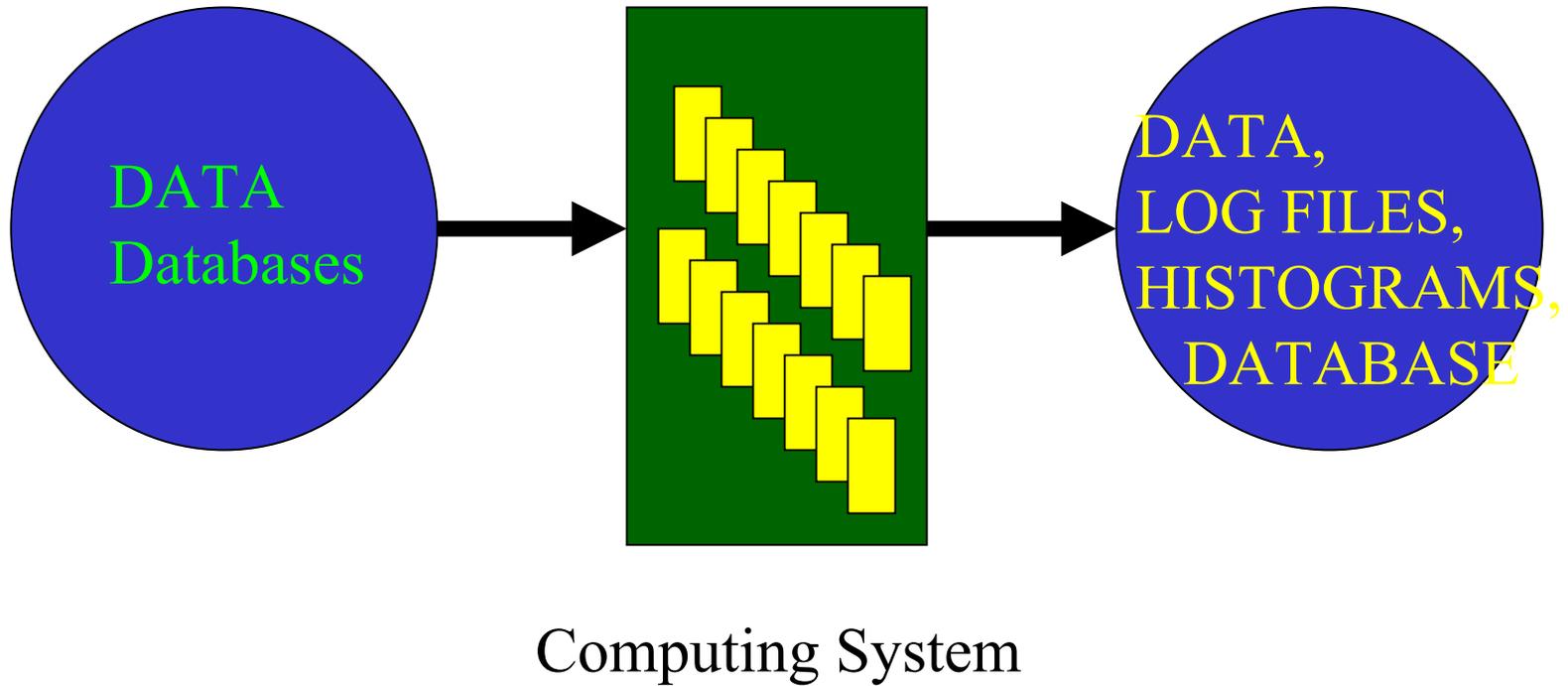
Reconstruction

Analysis

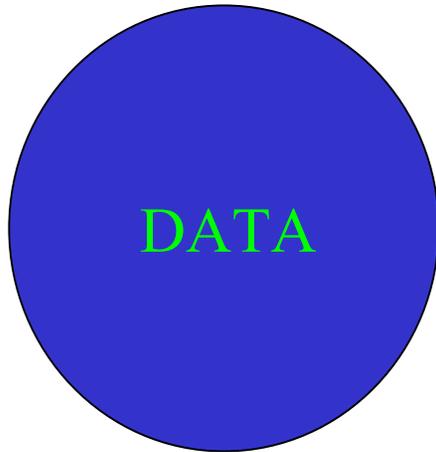
Simulation (Monte-Carlo)



# Distributed Computing Problem



# Distributed Computing Problem



How much data is there?

How is it organized?

In files?

How big are the files?

Within files?

By event? By object?

How big is an event  
or object?

How are they  
organized?

What kinds of data are there?

Event data?

Calibration data?

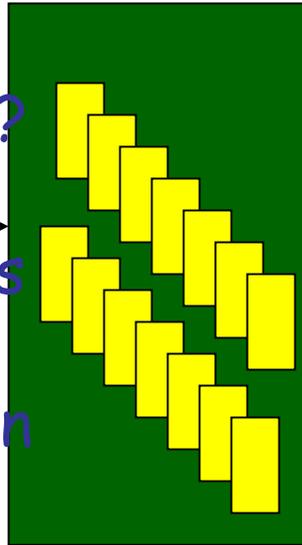
Parameters?

Triggers?



# Distributed Computing Problem

What is the system?  
How many systems?  
How are they connected?  
What is the bandwidth?  
How many data transfers  
can occur at once?  
What kind of information  
must be accessed?  
When?



What are the  
requirements for  
processing?  
Data flow?  
CPU?  
DB access?  
DB updates?  
O/P file  
updates?  
What is the goal  
for utilization?  
What is the  
latency  
desired?

What is the ratio of Computing System  
computation to data size?  
How are tasks scheduled?



## Distributed Computing Problem

How many files are there?

What type?

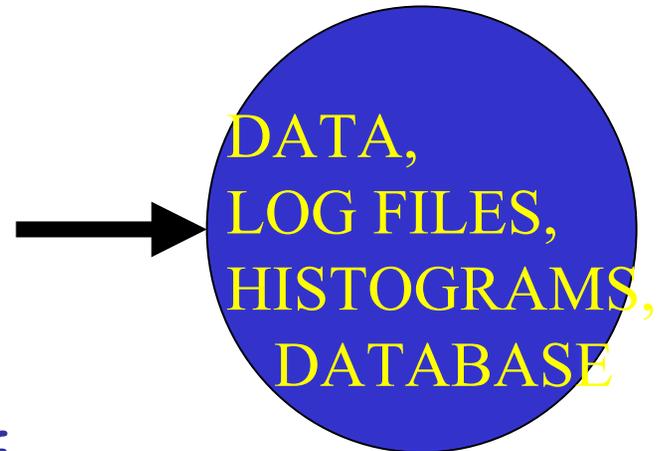
Where do they get written and archived?

How does one validate the production?

How is some data reprocessed if necessary?

Is there some priority scheme for saving results?

Do databases have to be updated?





## I: Level 3 or High Level Trigger

- **Characteristics:**
  - Huge CPU (CPU-limited in most cases)
  - Large Input Volume
  - Output/Input Volume ratio = 6-50
  - Moderate CPU/data
  - Moderate Executable size
  - Real-time system
  - Any mistakes lead to loss of data

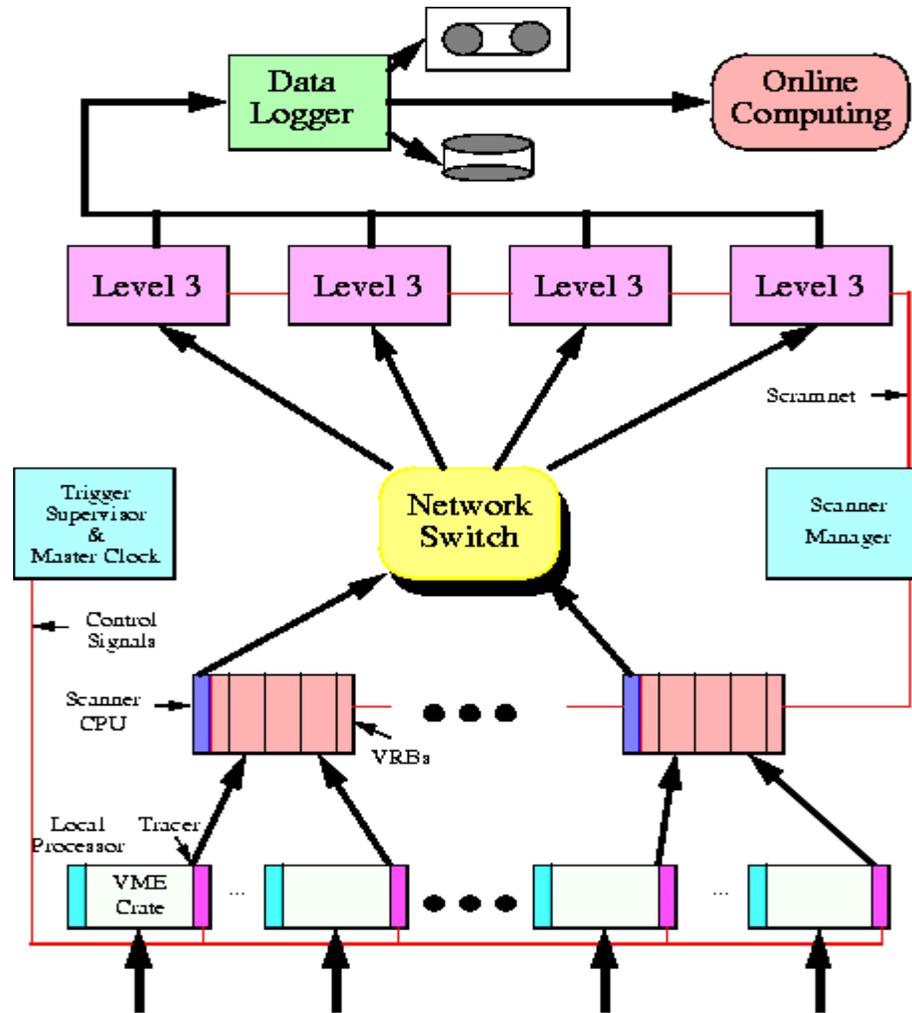


## Level 3

- Level 3 systems are part of the real-time data-taking of an experiment.
- But the system looks much like offline reconstruction:
  - Offline code is used
  - Offline framework
  - Calibrations are similar
  - Hardware looks very similar
- The output is the raw data of the experiment.



# Level 3 in CDF





# CMS Data Rates: From Detector to Storage

Physics filtering

40 MHz

~1000 TB/sec

Level 1 Trigger: Special Hardware

75 KHz

75 GB/sec

Level 2 Trigger: Commodity CPUs

5 KHz

5 GB/sec

Level 3 Trigger: Commodity CPUs

100 Hz

100 MB/sec

Raw Data to storage

Paul Avery

September, 2001

Stephen Wolbers, Heidi Schellman  
CERN School of Computing 2001

16



## Level 3 System Architecture

- Trigger Systems are part of the online and DAQ of an experiment.
- Design and specification are part of the detector construction.
- Integration with the online is critical.
- PCs and commodity switches are emerging as the standard L3 architecture.
- Details are driven by specific experiment needs.



## L3 Numbers

- **Input:**
  - CDF: 250 MB/s
  - CMS: 5 GB/s
- **Output:**
  - CDF: 20 MB/s
  - CMS: 100 MB/s
- **CPU:**
  - CDF: 10,000 SpecInt95
  - CMS: >440,000 SpecInt95 (not likely a final number)



## L3 Summary

- Large Input Volume
- Small Output/Input Ratio
  - Selection to keep only "interesting" events
- Large CPU, more would be better
- Fairly static system, only one "user"
- Commodity components (Ethernet, PCs, LINUX)



## II: Event Simulation (Monte Carlo)

- **Characteristics:**
  - Large total data volume.
  - Large total CPU.
  - Very Large CPU/data volume.
  - Large executable size.
  - Must be "tuned" to match the real performance of the detector/triggers, etc.
  - Production of samples can easily be distributed all over the world.



# Event Simulation Volumes

- Sizes are hard to predict but:
  - Many experiments and physics results are limited by "Monte Carlo Statistics".
  - Therefore, the number of events could increase in many (most?) cases and this would improve the physics result.
- **General Rule: Monte Carlo Statistics = 10 x Data Signal Statistics**
- **Expected:**
  - Run 2: 100's of TBytes
  - LHC: PBytes



## A digression: Instructions/byte, Spec, etc.

- Most HEP code scales with integer performance

- If:

- Processor A is rated at  $I_A$  "integer performance" and,
- Processor B is rated at  $I_B$
- Time to run on A is  $T_A$
- Time to run on B is  $T_B$

- Then:

- $T_B = (I_A/I_B) * T_A$



# SpecInt95, MIPS

- **SPEC:**

- SPEC is a non-profit corporation formed to establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers.

- **SPEC95:**

- Replaced Spec92, different benchmarks to reflect changes in chip architecture
- "A Sun SPARCstation 10/40 with 128 MB of memory was selected as the SPEC95 reference machine and Sun SC3.0.1 compilers were used to obtain reference timings on the new benchmarks. By definition, the SPECint95 and SPECfp95 numbers for the Sun SPARCstation 10/40 are both "1."
- One SpecInt95 is approximately 40 MIPS.
  - This is not exact, of course. We will use it as a rule of thumb.

- **SPEC2000**

- Replacement for Spec95, still not in common use.



# Event Simulation CPU

## Instructions/byte for event simulation:

- 50,000-100,000 and up.
- Depends on level of detail of simulation. Very sensitive to cutoff parameter values, among other things.

## Some examples:

- CDF:  $300 \text{ SI95-s} \cdot (40 \text{ MIP/SI95}) / 200 \text{ KB}$ 
  - 60,000 instructions/byte
- D0:  $3000 \text{ SI95} \cdot 40 / 1,200 \text{ KB}$ 
  - 100,000 instructions/byte
- CMS:  $8000 \text{ SI95} \cdot 40 / 2.4 \text{ MB}$ 
  - 133,000 instructions/byte
- ATLAS:  $3640 \text{ SI95} \cdot 40 / 2.5 \text{ MB}$ 
  - 58,240 inst./byte



## What do the instructions/byte numbers mean?

- Take a 1 GHz PIII
  - 48 SI95 (or about  $48 \times 40$  MIP)
- For a 50,000 inst./byte application
  - I/O rate:
    - $48 \times 40$  MIPS/50,000 inst/byte
    - = 38,400 byte/second
    - = 38 KB/s (very slow!)
    - Will take  $1,000,000/38 = 26,315$  seconds to generate a 1 GB file



## Event Simulation -- Infrastructure

- **Parameter Files**
- **Executables**
- **Calibrations**
- **Event Generators**
- **Particle fragmentation**
- **Etc.**



## Output of Event Simulation

- “Truth” - what the event really is, in terms of quark-level objects and in terms of hadronized objects and of hadronized objects after tracking through the detector.
- Objects (before and after hadronization)
  - Tracks, clusters, jets, etc.
  - Format: Ntuples, ROOT files, Objectivity, other.
- Histograms
- Log files
- Database Entries



## Summary of Event Simulation

- Large Output
- Large CPU
- Small (but important) input
- Easy to distribute generation
- Very important to “get it right” by using the proper specifications for the detector, efficiencies, interaction dynamics, decays, etc.



## III: Event Reconstruction

- **Characteristics:**
  - Large total data volume
  - Large total CPU
  - Large CPU/data volume
  - Large executable size
  - Pseudo real-time
  - Can be redone



# Event Reconstruction Volumes (Raw data input)

- **Run 2a Experiments**
  - 20 MB/s,  $10^7$  sec/year, each experiment
    - 200 Tbytes per year
- **RHIC**
  - 50-80 MB/s, sum of 4 experiments
    - Hundreds of Tbytes per year
- **LHC/Run 2b**
  - >100 MB/s,  $10^{**7}$  sec/year
    - >1 Pbyte/year/experiment
- **BaBaR**
  - >10 MB/s
    - >100 TB/year (350 TB so far)

# Event Reconstruction CPU



## • Instructions/byte for event reconstruction:

- CDF:  $100 \text{ SI95} \cdot 40 / 250 \text{ KB}$ 
  - 16,000 inst./byte
- D0:  $720 \text{ SI95} \cdot 40 / 250 \text{ KB}$ 
  - 115,000 instructions/byte
- CMS: 20,000 Million instructions/1,000,000 bytes
  - 20,000 instructions/byte (from CTP, 1997)
- CMS:  $3000 \text{ Specint95} / \text{event} \cdot 40 / 1 \text{ MB}$ 
  - 120,000 instructions/byte (2000 review)
- ATLAS:  $250 \text{ SI95} \cdot 40 / 1 \text{ MB}$ 
  - 10,000 instructions/byte (from CTP)
- ATLAS:  $640 \text{ SI95} \cdot 40 / 2 \text{ MB}$ 
  - 12,800 instructions/byte (2000 review)



# Instructions/byte for reconstruction

• CDF R1	15,000	}	Fermilab Run 1, 1995
• D0 R1	25,000		
• E687	15,000	}	Fermilab FT, 1990-97
• E831	50,000		
• CDF R2	16,000	}	Fermilab Run 2, 2001
• D0 R2	64,000		
• BABAR	75,000		
• CMS	20,000 (1997 est.)		
• CMS	120,000 (2000 est.)		
• ATLAS	10,000 (1997 est.)		
• ATLAS	12,800 (2000 est.)		
• ALICE	160,000 (pb-pb)		
• ALICE	16,000 (p-p)		
• LHCb	80,000		



# Output of Event Reconstruction

- **Objects**
  - Tracks, clusters, jets, etc.
  - Format: Ntuples, ROOT files, DSPACK, Objectivity, other.
- **Histograms and other monitoring information**
- **Log files**
- **Database Entries**



## Summary of Event Reconstruction

- Event Reconstruction has large input, large output and large CPU/data.
- It is normally accomplished on a farm which is designed and built to handle this particular kind of computing.
- Nevertheless, it takes effort to properly design, test and build such a farm (see Lecture 2).



## IV: Event Selection and Secondary Datasets

- Smaller datasets, rich in useful events, are commonly created.
- The input to this process is the output of reconstruction.
- The output is a much-reduced dataset to be read many times.
- The format of the output is defined by the experiment.



## Secondary Datasets

- Sometimes called DSTs, PADs, AODs, NTUPLES, etc.
- Each dataset is as small as possible to make analysis as quick and efficient as possible.
- However, there are competing requirements for the datasets:
  - Smaller is better for access speed, ability to keep datasets on disk, etc.
  - More information is better if one wants to avoid going back to raw or reconstruction output to refit tracks, reapply calibrations, etc.
- An optimal size is chosen for each experiment and physics group to allow for the most effective analysis.



## Producing Secondary Datasets

- **Characteristics:**
  - CPU: Depends on input data size.
  - Instructions/byte: Ranges from quite small (event selection using small number of quantities) to reasonably large (unpack data, calculate quantities, make cuts, reformat data).
  - Data Volume: Small to Large.
    - $\text{Sum}_{\text{All Sets}} = 33\%$  of Raw data (CDF)
      - Each set is approx. a few percent



## Summary of Secondary Dataset Production

- Not a well-specified problem.
- Sometimes I/O bound, sometimes CPU bound.
- Number of “users” is much larger than Event Reconstruction.
- Computing system needs to be flexible enough to handle these specifications.



## V: Analysis of Final Datasets

- Final Analysis is characterized by:
  - (Not necessarily) small datasets.
  - Little or no output, except for NTUPLES, histograms, fits, etc.
  - Multiple passes, interactive.
  - Unpredictable input datasets.
    - Driven by physics, corrections, etc.
  - Many, many individuals.
  - Many, many computers.
  - Relatively small instructions/byte.
  - $\text{Sum}_{\text{All Activity}} = \text{Large (CPU, IO, Datasets)}$

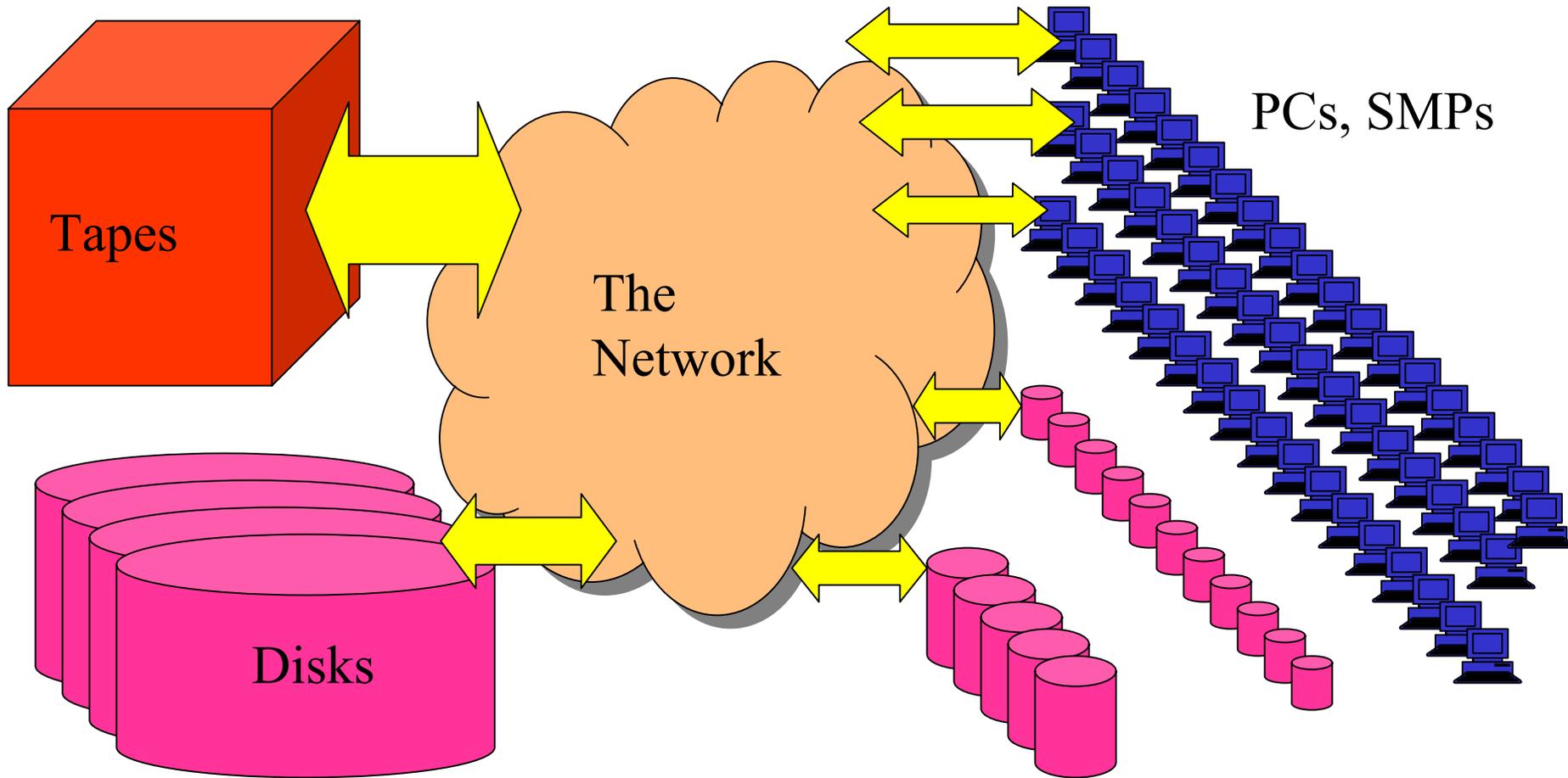


# Data analysis in international collaborations: past

- In the past analysis was centered at the experimental sites
  - a few major external centers were used.
  - Up the mid 90s bulk data were transferred by shipping tapes, networks were used for programs and conditions data.
  - External analysis centers served the local/national users only.
  - Often staff (and equipment) from the external center being placed at the experimental site to ensure the flow of tapes.
  - The external analysis often was significantly disconnected from the collaboration mainstream.



# Analysis - a very general model



September, 2001

Stephen Wolbers, Heidi Schellman  
CERN School of Computing 2001

41



## Some Real-Life Analysis Systems

- **Run 2**
  - **DO: Central SMP + Many LINUX boxes**
    - Issues: Data Access, Code Build time, CPU required, etc.
    - Goal: Get data to people who need it quickly and efficiently
    - Data stored on tape in robots, accessed via a software layer (SAM)



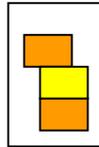
# Data Tiers for a single Event (D0)

~200B



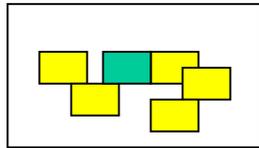
Data Catalog entry

5-15KB



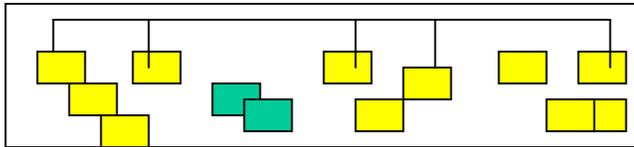
Condensed summary physics data

50-100KB



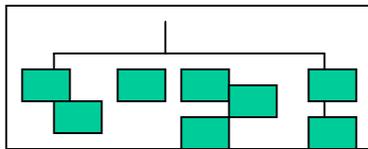
Summary Physics Objects

~350KB



Reconstructed Data -  
Hits, Tracks, Clusters, Particles

250KB



RAW detector measurements



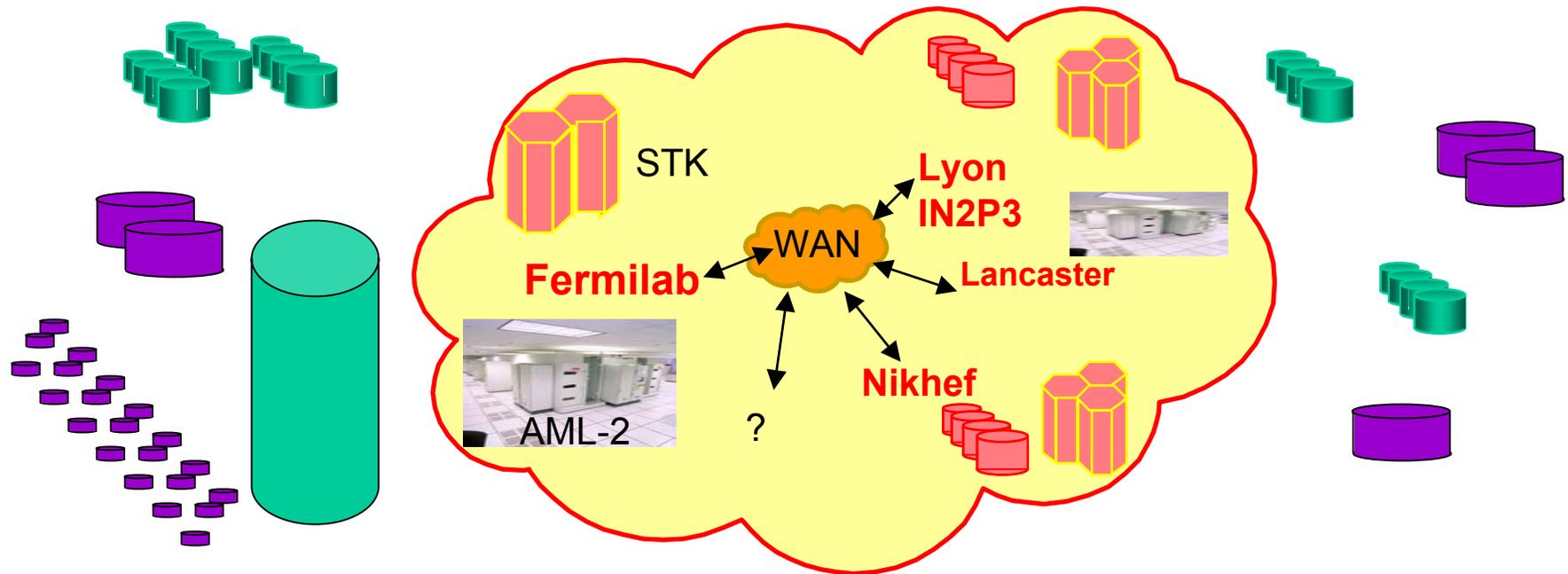
# D0 Fully Distributed Network-centric Data Handling System

- D0 designed a distributed system from the outset
- D0 took a different/orthogonal approach to CDF
  - Network-attached tapes (via a Mass Storage System)
  - Locally accessible disk caches
- The data handling system is working and installed at 13 different 'Stations' - 6 at Fermilab, 5 in Europe and 2 in US (plus several test installations)



# The Data Store and Disk Caches

Data Store stores read-only Files on permanent tape or disk storage

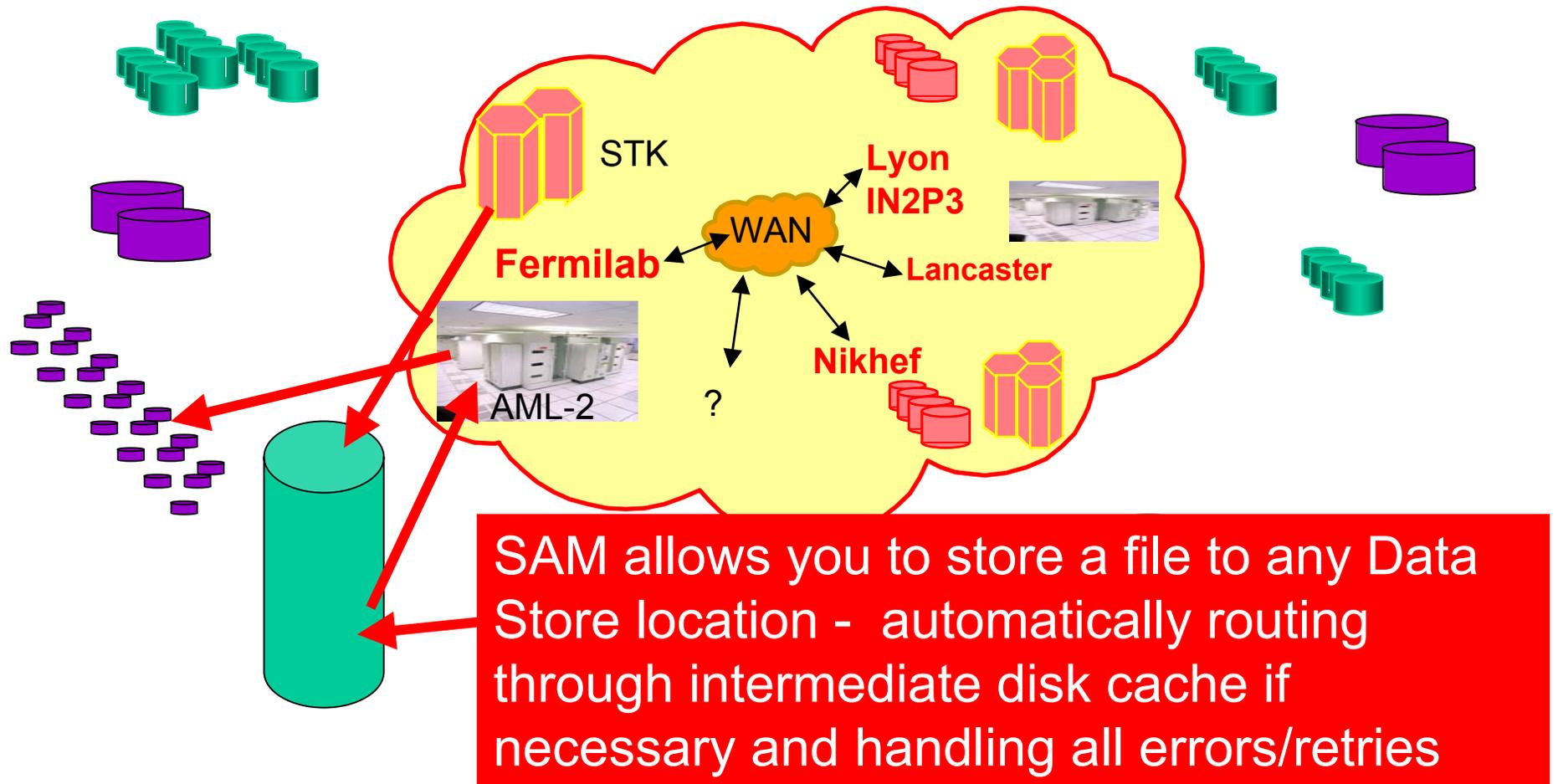


All processing jobs read sequentially from locally attached disk cache.  
“**S**equential **A**ccess through **M**etadata” – **SAM**  
Input to all processing jobs is a Dataset

Event level access is built on top of file level access using catalog/index



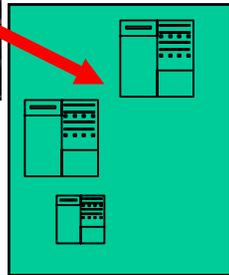
# The Data Store and Disk Caches



# SAM Processing Stations at Fermilab



"data-logger"



12-20 MBps

"d0-test" and  
"sam-cluster"

"central-analysis"

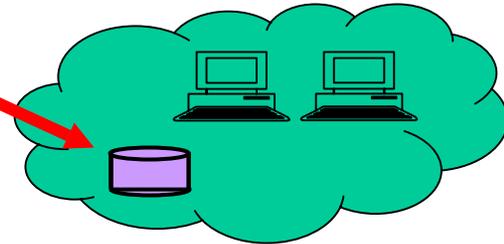


100+ MBps

400+ MBps



Enstore  
Mass Storage System

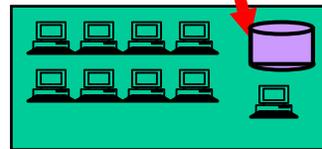
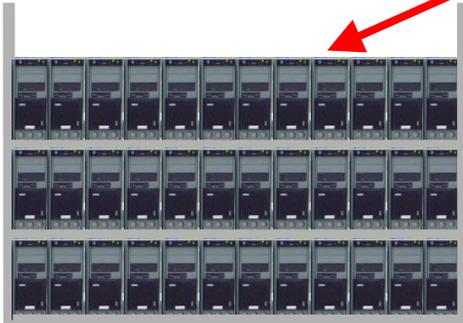


"linux-analysis-clusters"

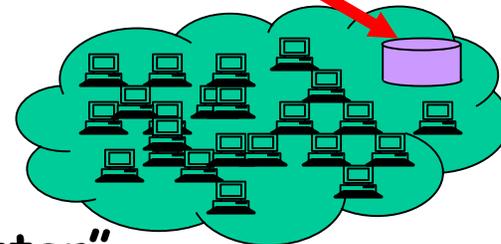
"farm"



12-20 MBps



"linux-build-cluster"



"clueD0"  
~100  
desktops

September, 2001

Stephen Wolbers, Heidi Schellman  
CERN School of Computing 2001



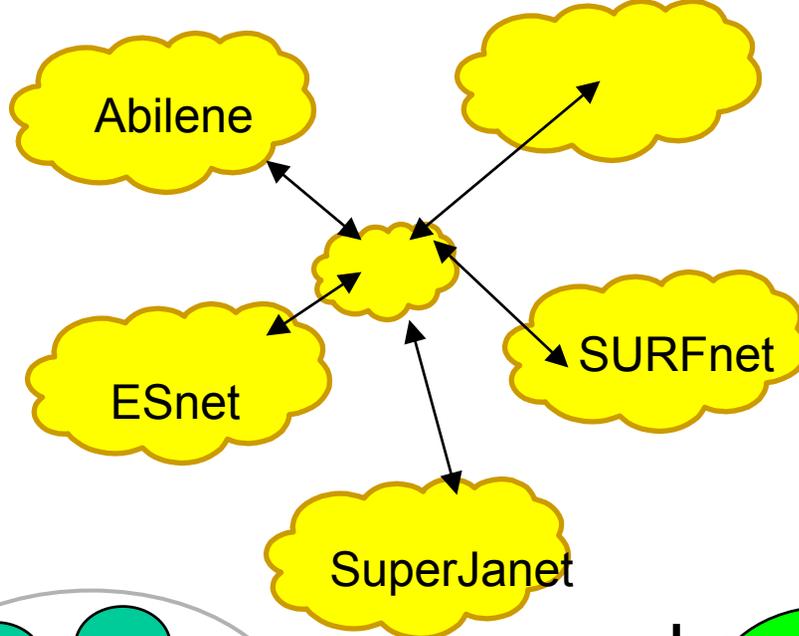
# DO Processing Stations Worldwide

● = MC production centers (#nodes all duals)

Lyon/IN2P3

100

MSU

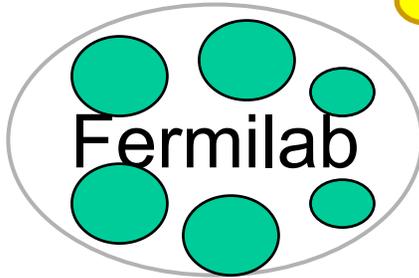


Prague  
32

Columbia

NIKHEF  
50

UTA  
64



Lancaster  
200

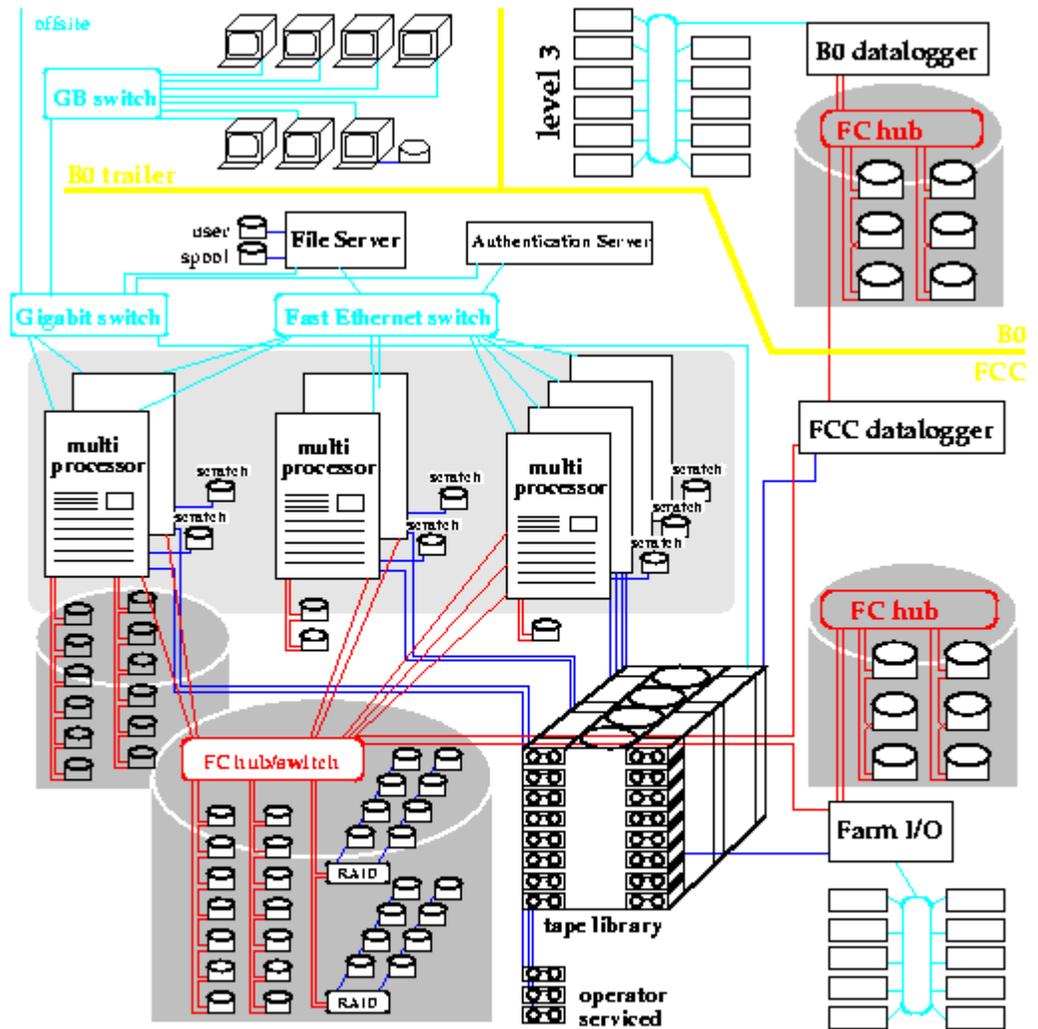
Imperial  
College

# Data Access Model: CDF



- Ingredients:**

- Gigabit Ethernet
- Raw data are stored in tape robot located in FCC
- Multi-CPU analysis machine
- High tape access bandwidth
- Fiber Channel connected disks



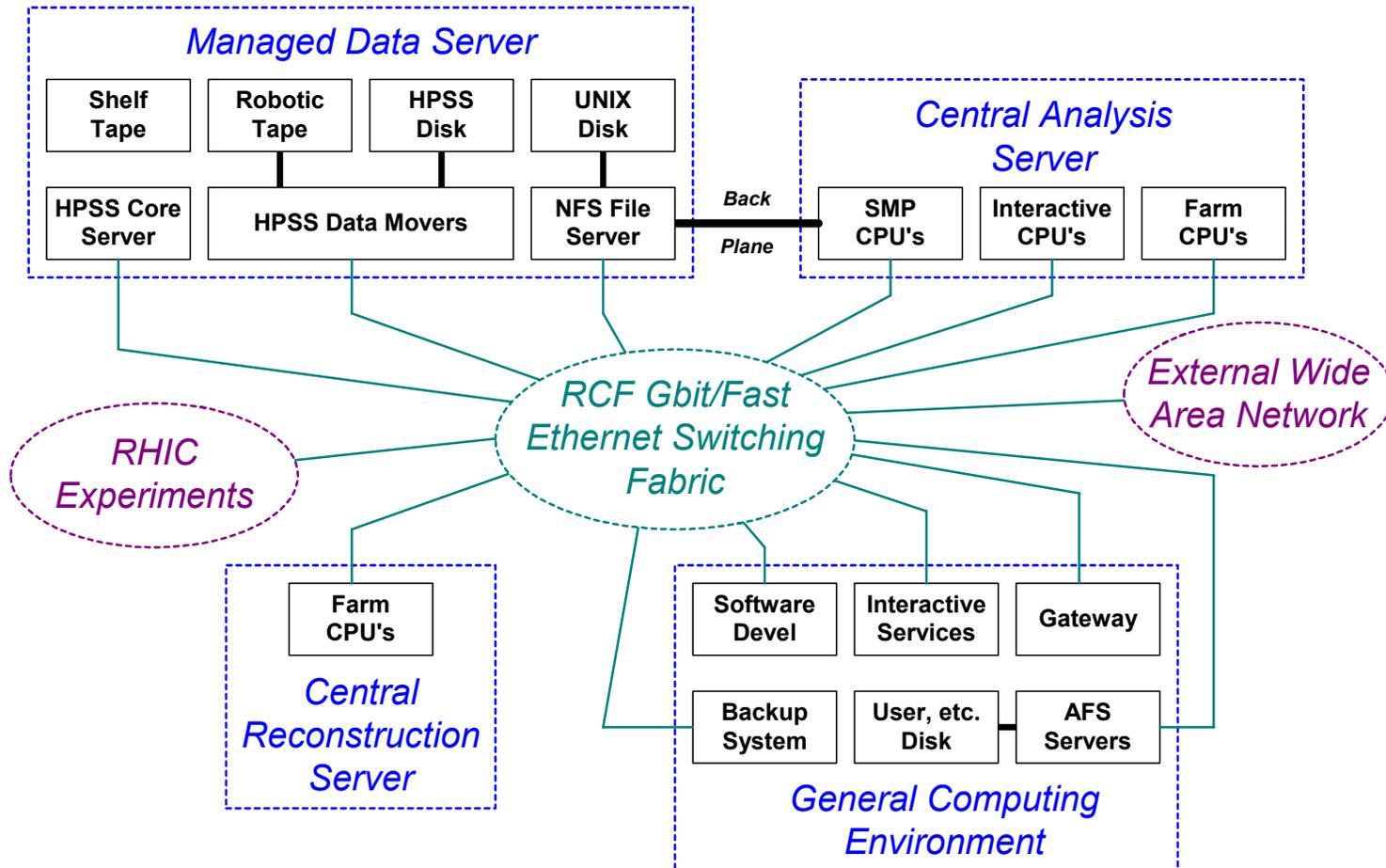
# Computing Model for Run 2a

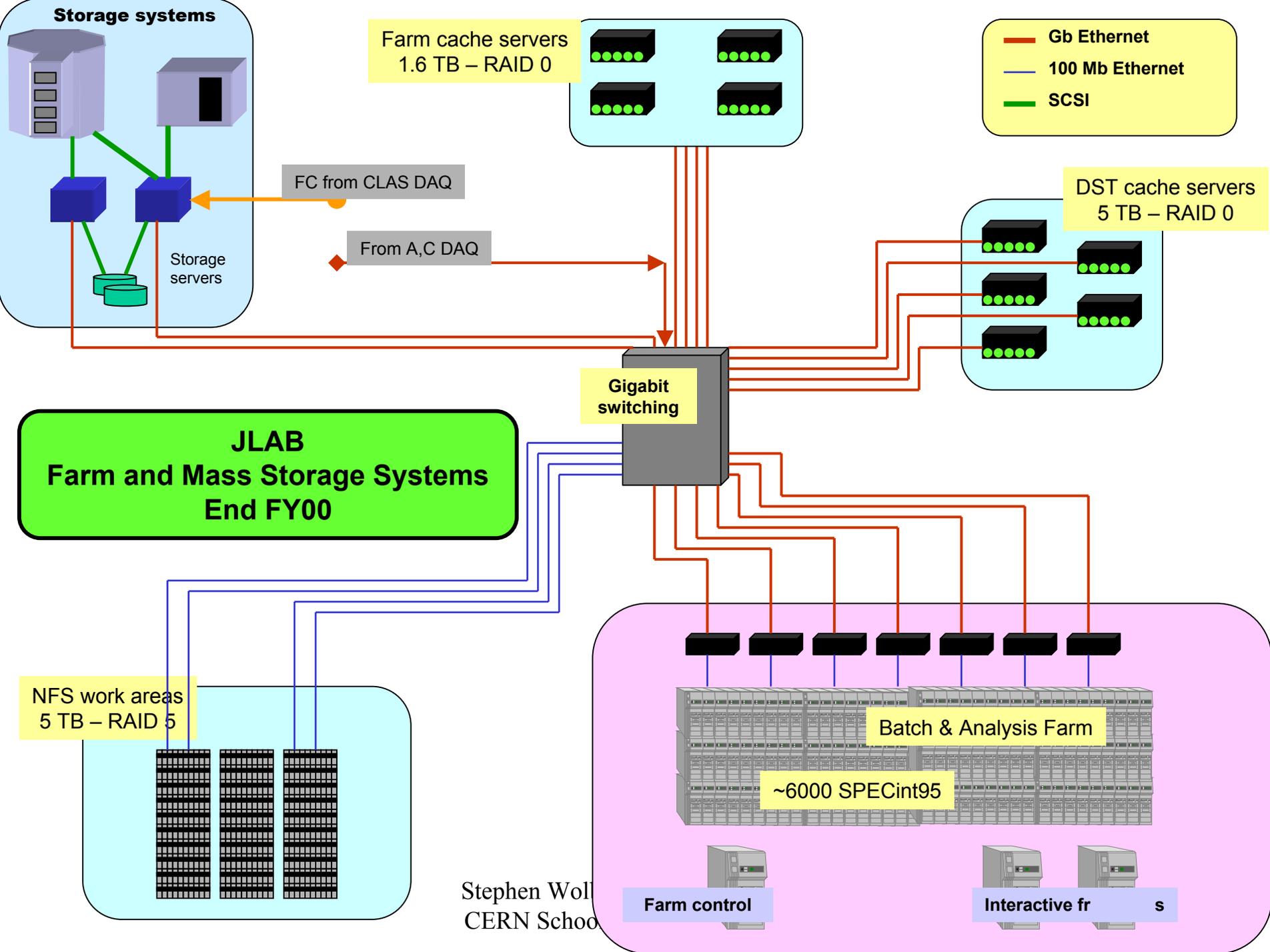


- CDF and D0 have similar but not identical computing models.
  - In both cases data is logged to tape stored in large robotic libraries.
  - Event reconstruction is performed on large Linux PC farms.
  - Analysis is performed on medium to large multi-processor computers
  - Final analysis, paper preparation, etc. is performed on Linux desktops or Windows desktops.



# RHIC Computing Facility







# BaBar: Worldwide Collaboration of 80 Institutes

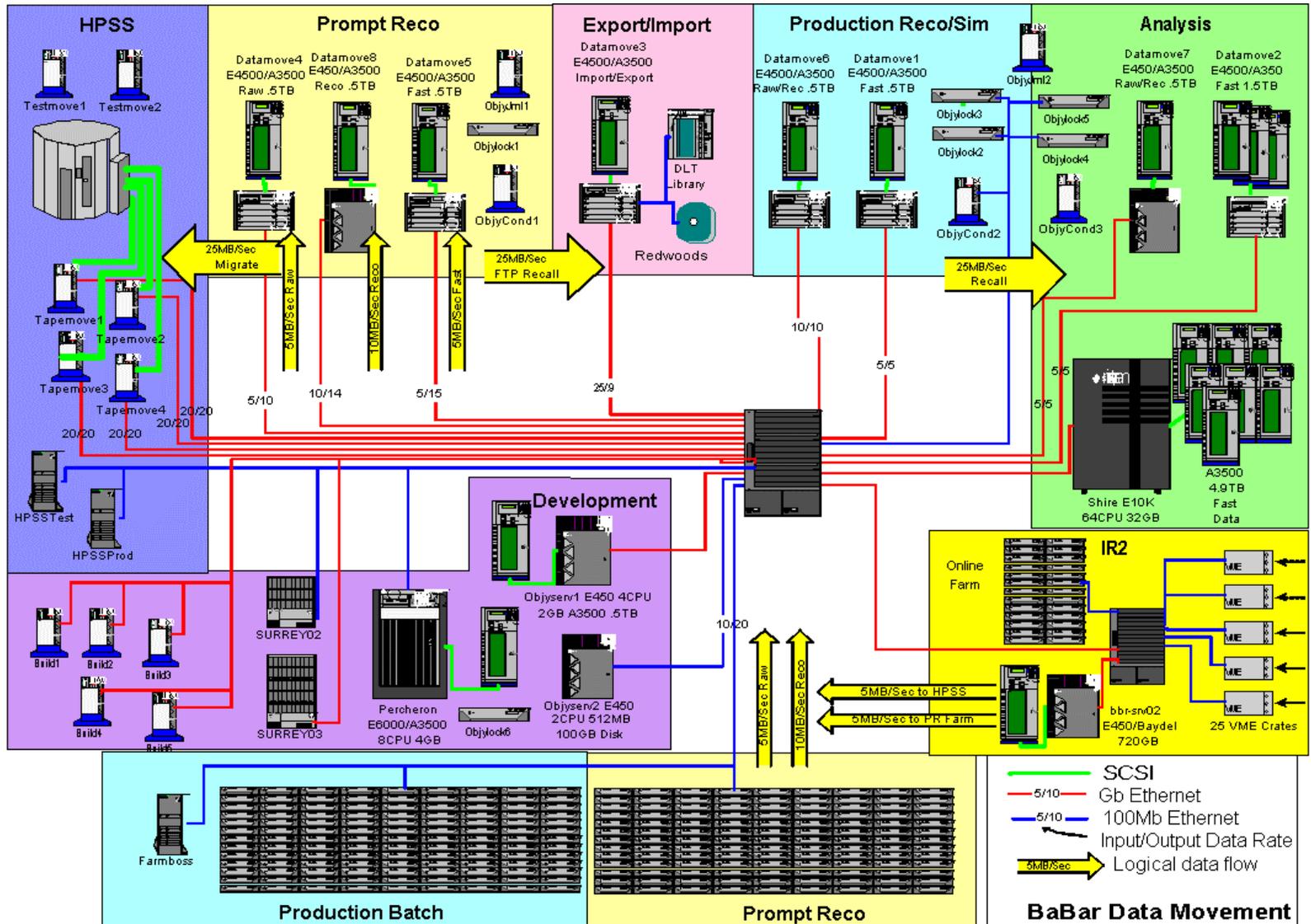


September, 2001

Stephen Wolbers, Heidi Schellman  
CERN School of Computing 2001



# BaBar Offline Systems: August 1999





## Putting it all together

- **A High-Performance Distributed Computing System consists of many pieces:**
  - High-Performance Networking
  - Data Storage and access ("tapes")
  - Central CPU+Disk Resources
  - Distributed CPU+Disk Resources
  - Software Systems to tie it all together, allocate resources, prioritize, etc.



## Summary of Lecture I

- Analysis of the problem to be solved is important.
- Issues such as data size, file size, CPU, data location, data movement, all need to be examined when analyzing computing problems in High Energy Physics.
- Solutions depend on the analysis and will be explored in Lecture II.