CDF Level 3 Trigger Online Cluster

J. Tseng
Massachusetts Institute of Technology

LCCWS
May 24, 2001

- Online Requirements
- Cluster Description
- Real-time Operation
- Conclusion

Online Requirements

mainframes → clusters: online as well

▷ Core application: event filtering
(select interesting events from stream)

▷ Input from DAQ: ATM switch

- 16 × 155Mbps ports ⇒ max 260 MB/s
- event size ∼ 250 KB

▷ Output: max data logging rate 20 MB/s

▷ Spec: $>$ 300 ev/s input, $<$ 75 ev/s output

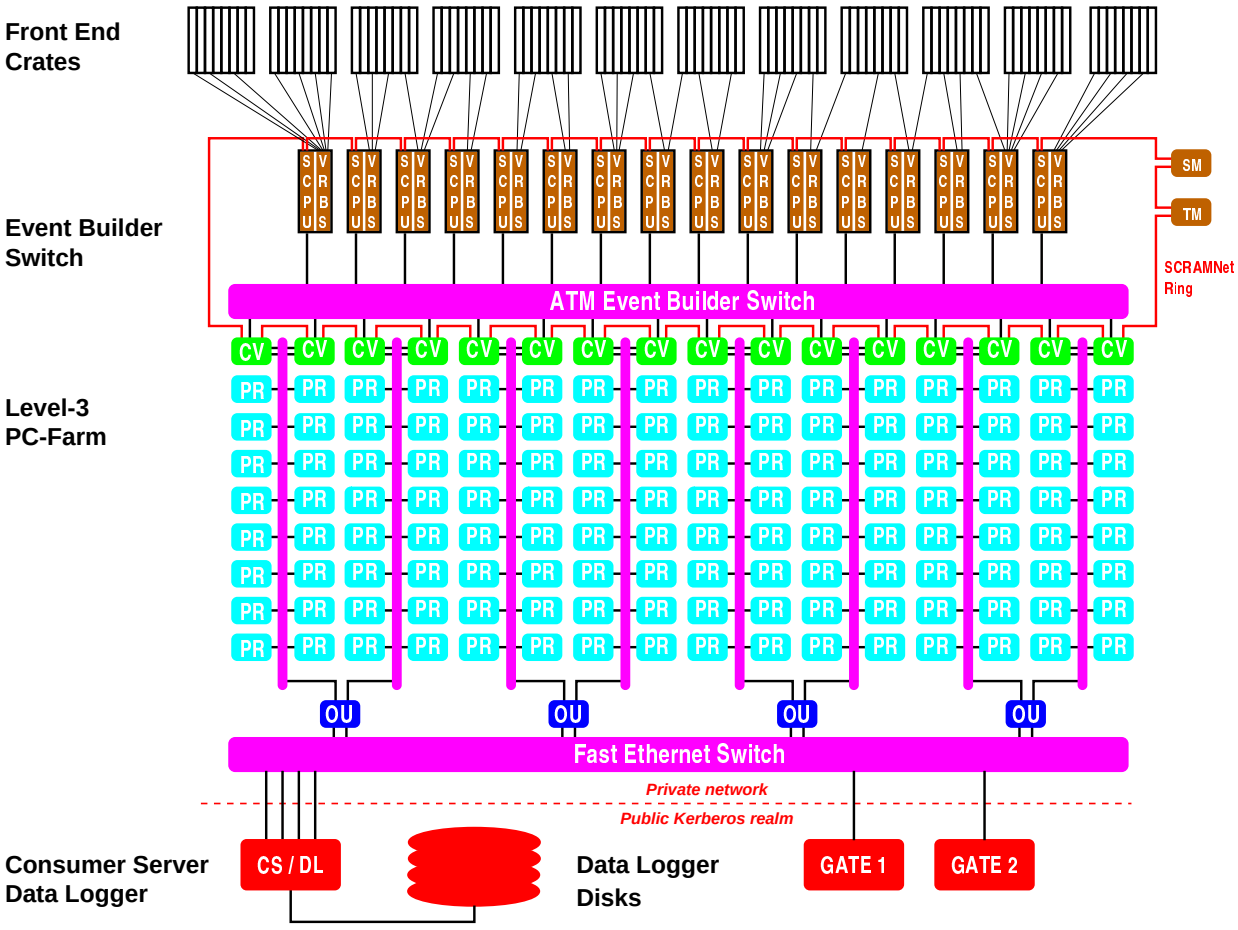
▷ Long uptimes (data waits for no one)

▷ Start/stop times: ∼ 5 minutes

▷ High-throughput error monitoring
(data corruption detection)

System in use now
Run 2 already begun

Level 3 Farm



▷ 16 "converters": ATM in, FastEthernet out

▷ 4 output nodes > 20 MB/s

▷ Buffer data only in memory

▷ LAN traffic control crucial to performance

⇒ private network, 2 gateways

Cluster Hardware

- Computers:
  - acquired in stages since 1998
  - dual Pentium II/III
    400 to 850 MHz, 128 to 256 MB
  - present: Linux 2.2.14 (2.0.38 on cv)
    FNAL-supported, local NFS install
  - target $2000/node (actual usually less)
  - 149 nodes in cluster
- Switches:
  - 3Com SuperStack 24-port Fast Ethernet
  - central bank of 9 switches
- Packaging:
  - commodity cases and shelves

Cluster Hardware (II)

B0 3rd Floor Computing Room

Real-time Operation

Downtimes determined by collider, not cluster

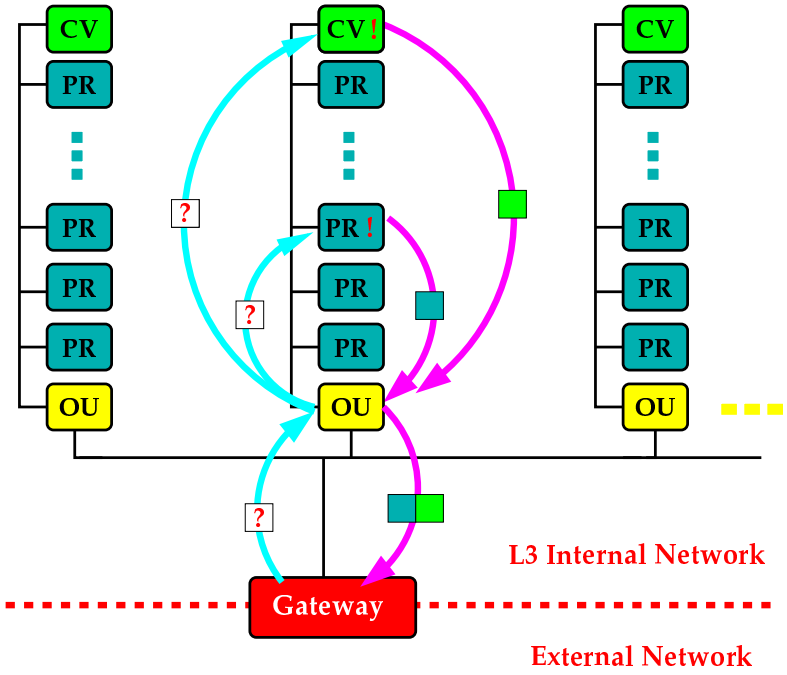
- ▷ Interchangeable nodes and components
- ▷ Each node tested relative to purchase specification when received, *e.g.*,
  - $\surd$ FNAL `tiny` performance benchmark
  - $\surd$ CPU temperature under load
  - $\surd$ Disk I/O rate
- ▷ Filter developed/tested entirely offline

Initialization

Online operation requires rapid turnaround between runs

▷ Control of 149 nodes must be parallel

⇒ Parallel CORBA operations



Control Interface

ROOT/CINT for control software:

- interpreted C++: OO command-line (expert) interface
- classes describe nodes as well as aggregates
- same classes used by Run Control interface
- output caught by ROOT classes

$\Rightarrow$ very convenient for tool building

File Distribution/Collection

▷ Offline environment duplicated for filter on all nodes in private network

- Filter executable and libraries ($\sim$ 100 MB)
- Trigger table and databases ($\sim$ 100 MB)
- Calibrations change every run

▷ Regularly distribute $\sim$ 100 MB files in $\sim$ 5 minute turnaround

$\Rightarrow$ special pipeline program
(like `ptcp` without MPI)

▷ Automated core/log file collection for crashed filters

- Run Control-specified maximum
- Packaged and uploaded for experts

Real-time Monitoring

▷ Periodic status reports (every 4 sec/node)

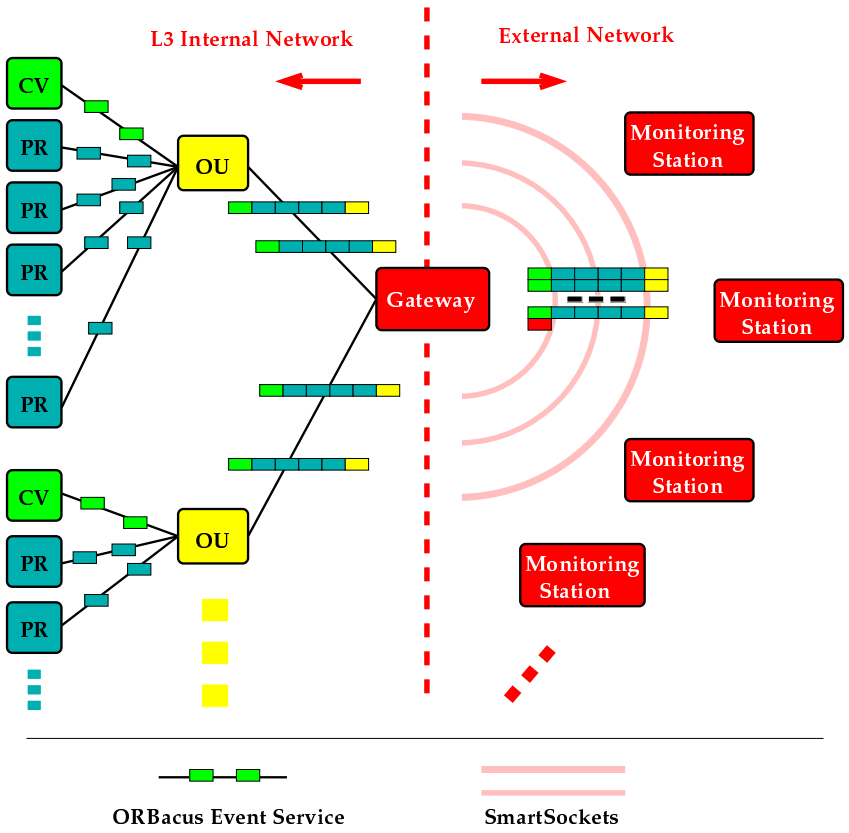
▷ Reports error messages

Upstream data corruption ⇒ lots of messages

√ Must be robust for (common) condition

√ Prescaled report rate ⇒ bursts

√ Message collation inside farm



Conclusion

▷ CDF Level 3: 149 dual-CPU PC farm

▷ Total $\sim$ 250 nodes by end of 2001

▷ Core application: real-time event filtering

▷ Achieving real-time operation:

- More computing/event: expandable sub-farms (input/output fixed)
- Rapid control: parallel CORBA operations
- Large file distribution: pipelined copy
- High-throughout error reporting: message collation

System in use now
Run 2 already begun