The Architecture of the ZEUS Second Level Global Tracking Trigger

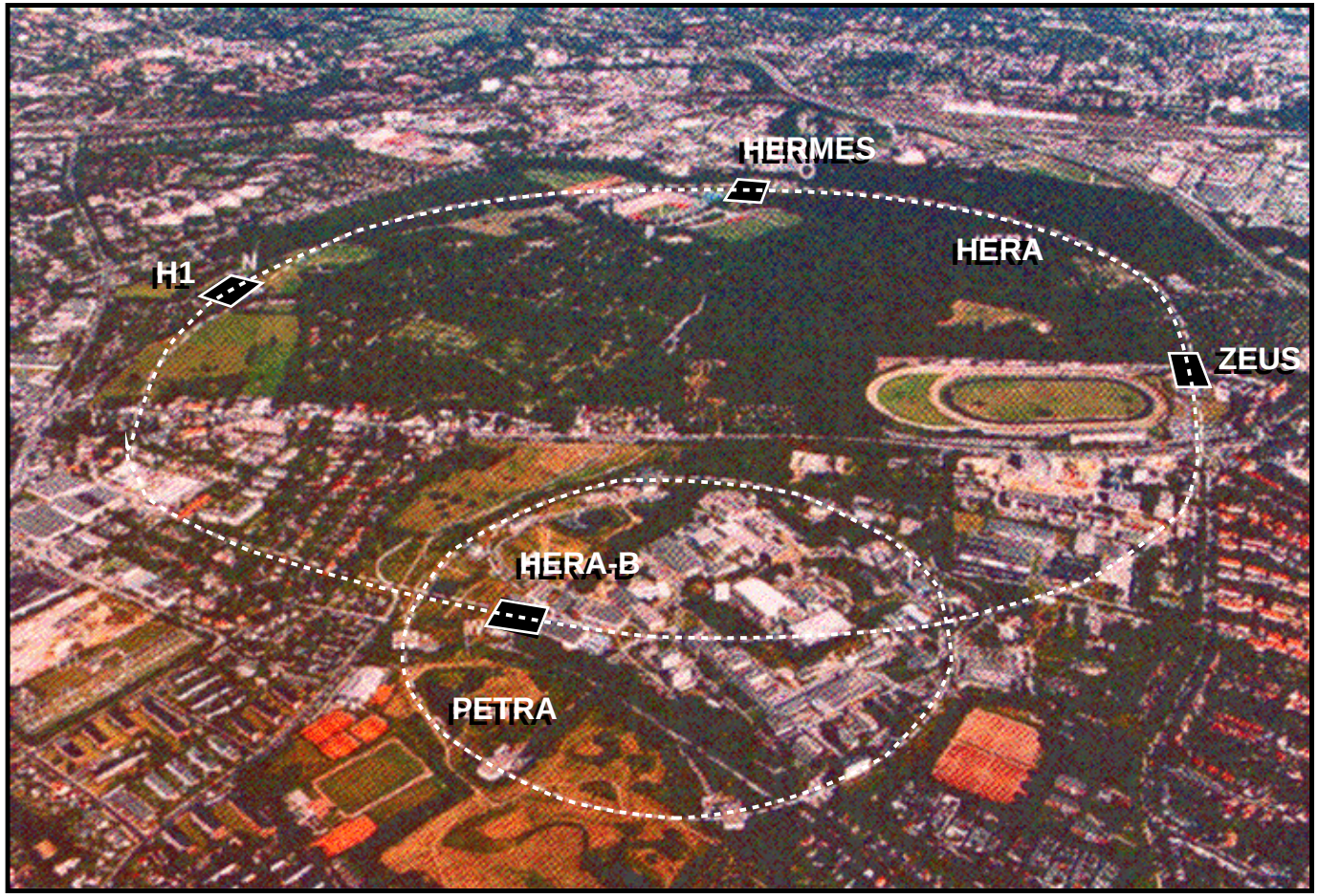
Mark Sutton
University College London

27th September 2004

On behalf of the ZEUS GTT group:

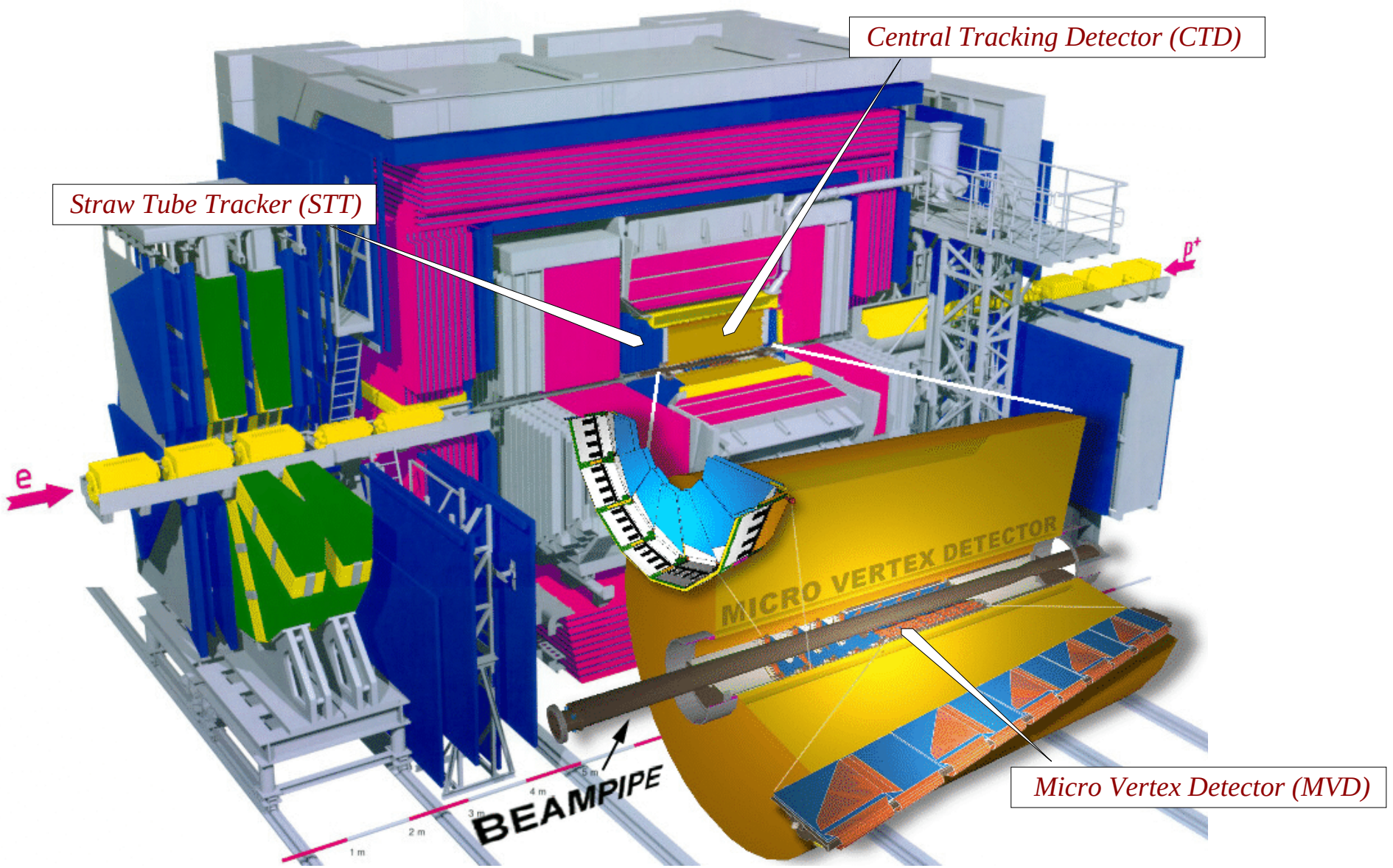
DESY-Hamburg, MEPHI, University of Glasgow, University of Oxford
University College London, Yale

The HERA Collider

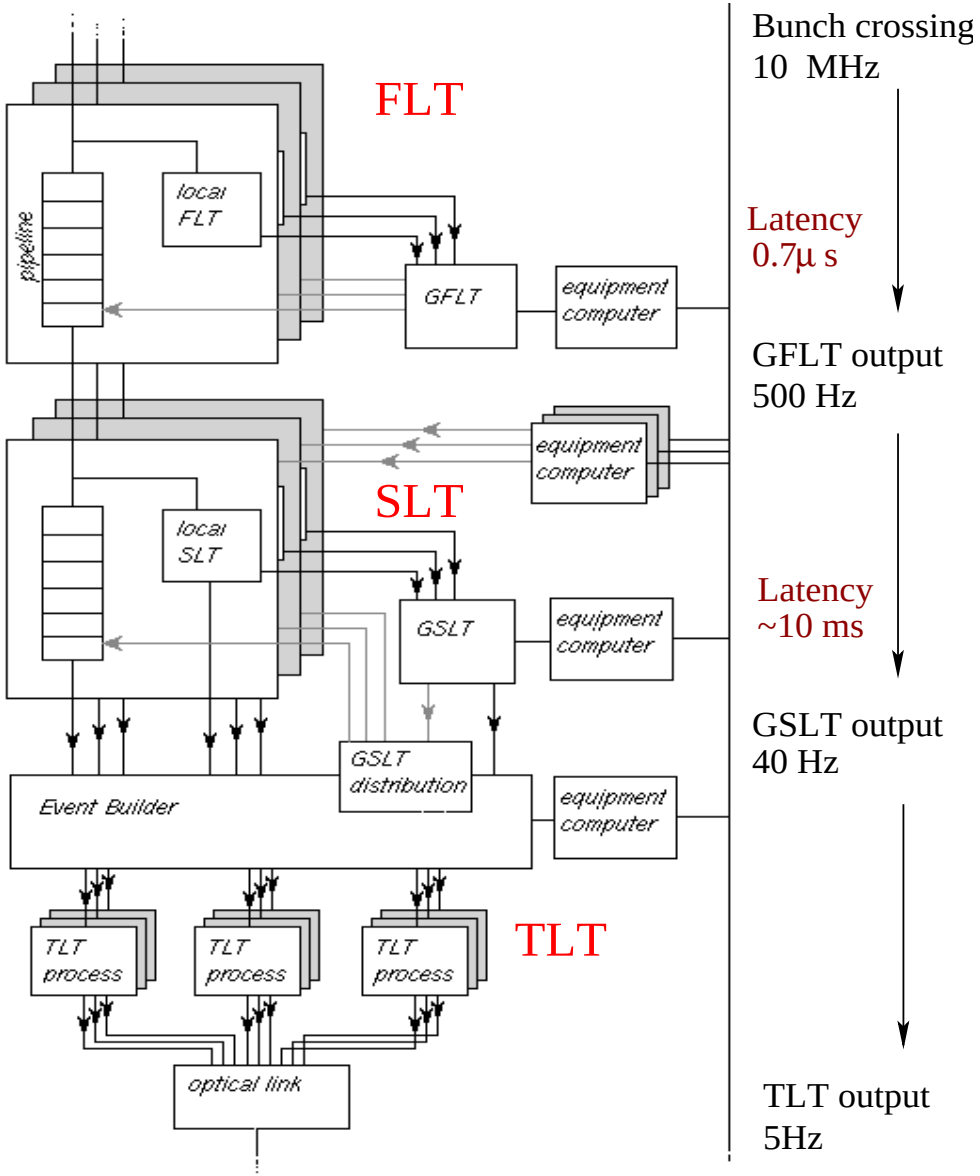


- HERA is a lepton-hadron colliding beam machine.
- 27.5 GeV leptons on 920 GeV protons, bunch crossing interval 96 ns.

The ZEUS Detector



The ZEUS Trigger



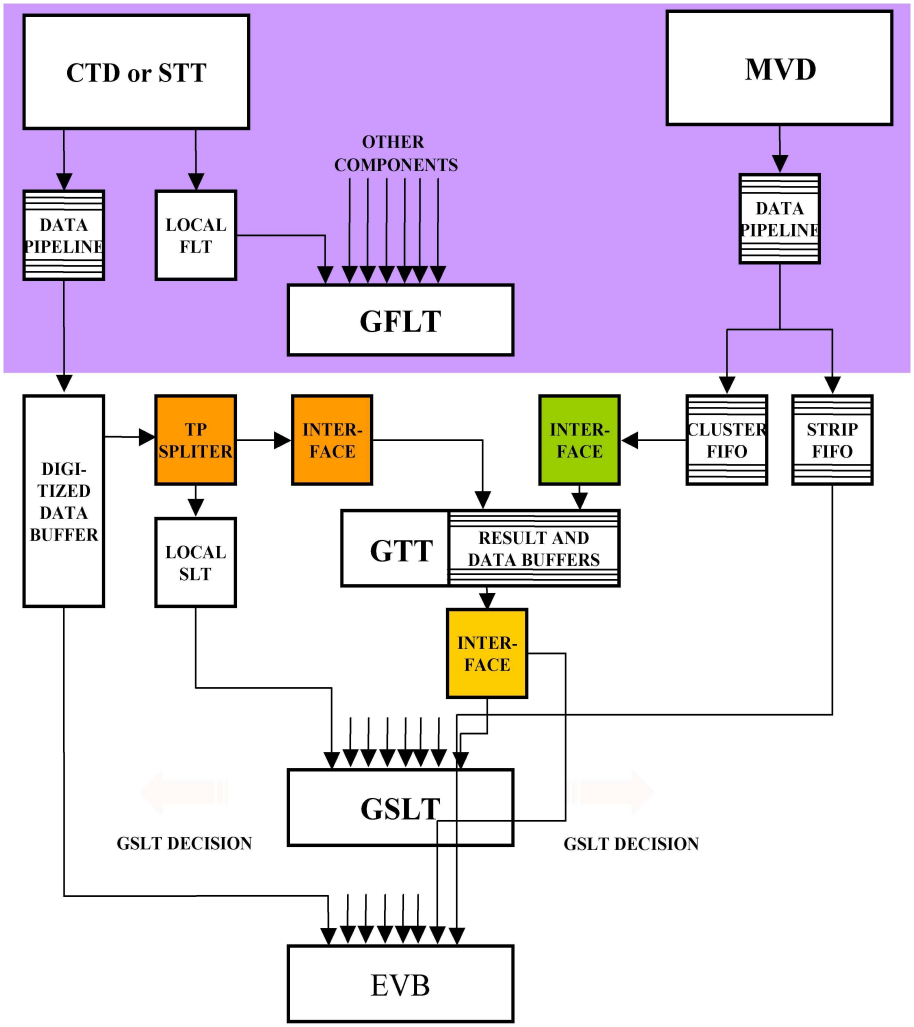
- Built in 1992 ⇒ First High Rate, pipelined trigger – 96 ns bunch crossing.
- Flexible three level trigger.
- First Level Trigger (FLT) deadtime free.
- Second Level Trigger (SLT) based on INMOS Transputers (20 Mbit/s data transfer rate).

Introduction

- HERA luminosity Upgrade ⇛ ZEUS added MVD and forward STT.
- Before upgrade, at Second Level Trigger, tracking only with CTD
  - ▷ Transputer technology more than 10 years old, not easily adaptable.
- New trigger must deliver trigger result within $\sim$ 15 ms.
- Improved commodity CPU and network technology make better, faster tracking possible, ⇛ combined tracking with CTD, MVD and STT data
  - ▷ Extend to forward region.
  - ▷ Improve vertex resolution, ⇛ eventually allow heavy flavour tagging online in the Second Level Trigger.

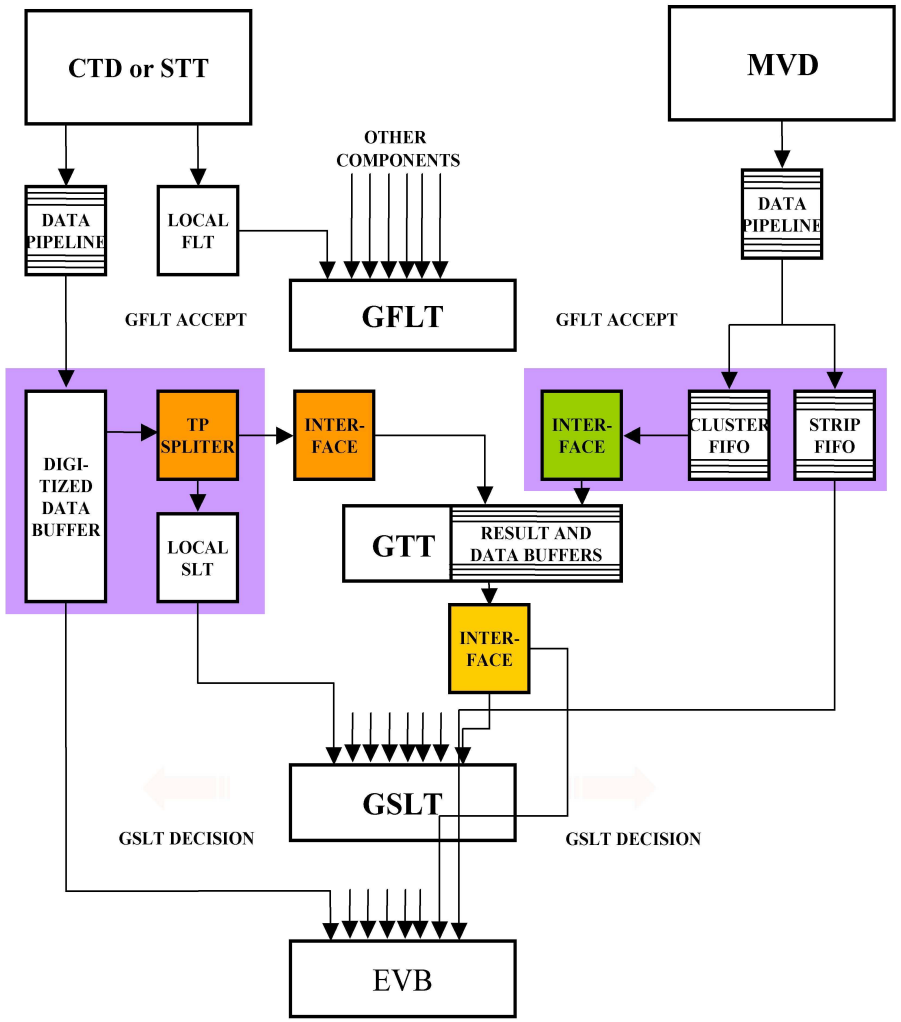
Overview of the GTT system

- Detector data pipelined at bunch crossing (10MHz). On GFLT accept..
- Transputer based readout of CTD and STT (∼3 ms). Within the Transputer system data is forked and sent to:
  - ▷ GTT component interface,
  - ▷ Existing component SLT. On a GSLT accept also to the Event Builder.
- MVD-GTT component interface integrated with MVD VME readout (∼1 ms).
- Component VME interfaces ship component raw data to GTT 1GHz Linux PC farm.
- Algorithm result banks sent to Global Second Level Trigger (GSLT).



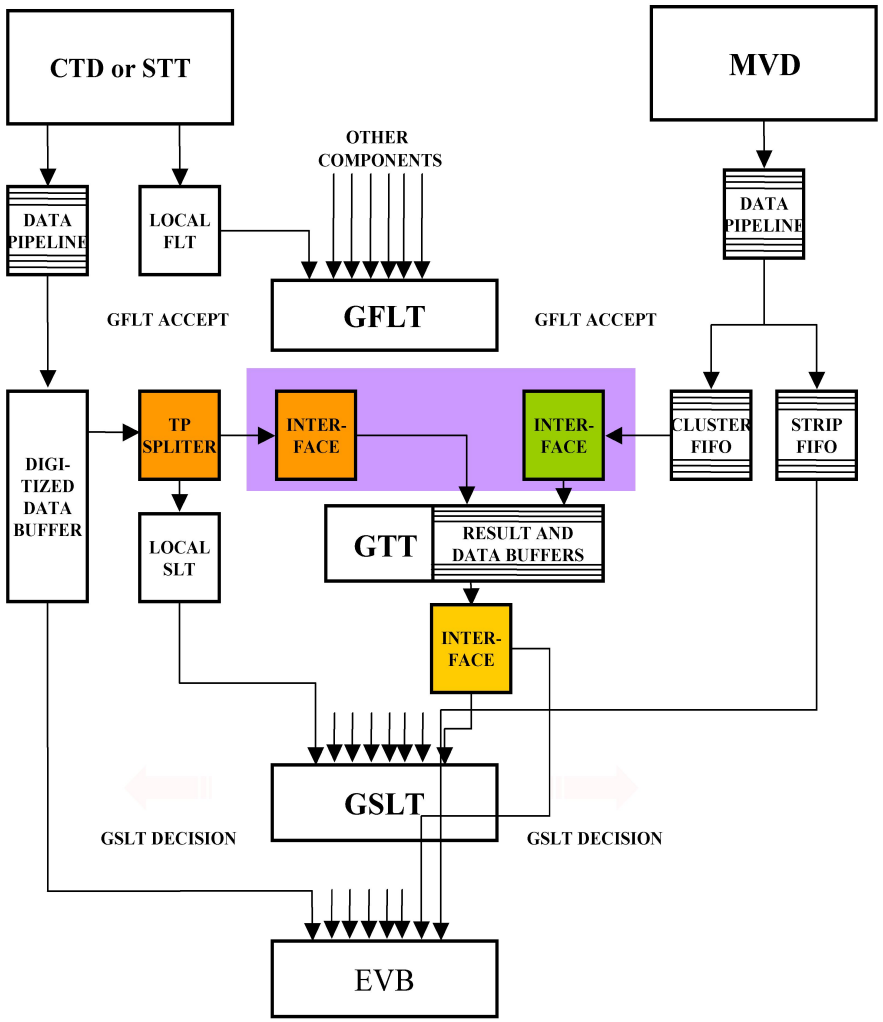
Overview of the GTT system

- Detector data pipelined at bunch crossing (10MHz). On GFLT accept..
- Transputer based readout of CTD and STT (∼3 ms). Within the Transputer system data is forked and sent to:
  - ▷ GTT component interface,
  - ▷ Existing component SLT. On a GSLT accept also to the Event Builder.
- MVD-GTT component interface integrated with MVD VME readout (∼1 ms).
- Component VME interfaces ship component raw data to GTT 1GHz Linux PC farm.
- Algorithm result banks sent to Global Second Level Trigger (GSLT).



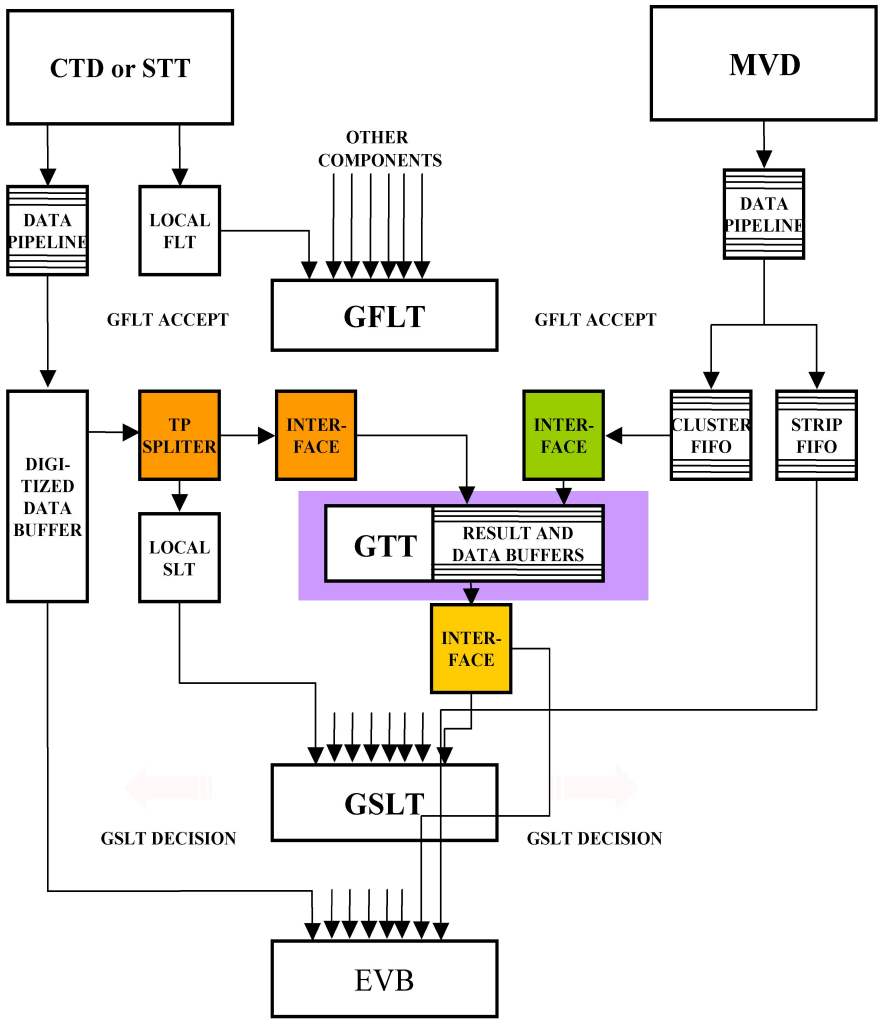
Overview of the GTT system

- Detector data pipelined at bunch crossing (10MHz). On GFLT accept..
- Transputer based readout of CTD and STT (∼3 ms). Within the Transputer system data is forked and sent to:
  - ▷ GTT component interface,
  - ▷ Existing component SLT. On a GSLT accept also to the Event Builder.
- MVD-GTT component interface integrated with MVD VME readout (∼1 ms).
- Component VME interfaces ship component raw data to GTT 1GHz Linux PC farm.
- Algorithm result banks sent to Global Second Level Trigger (GSLT).



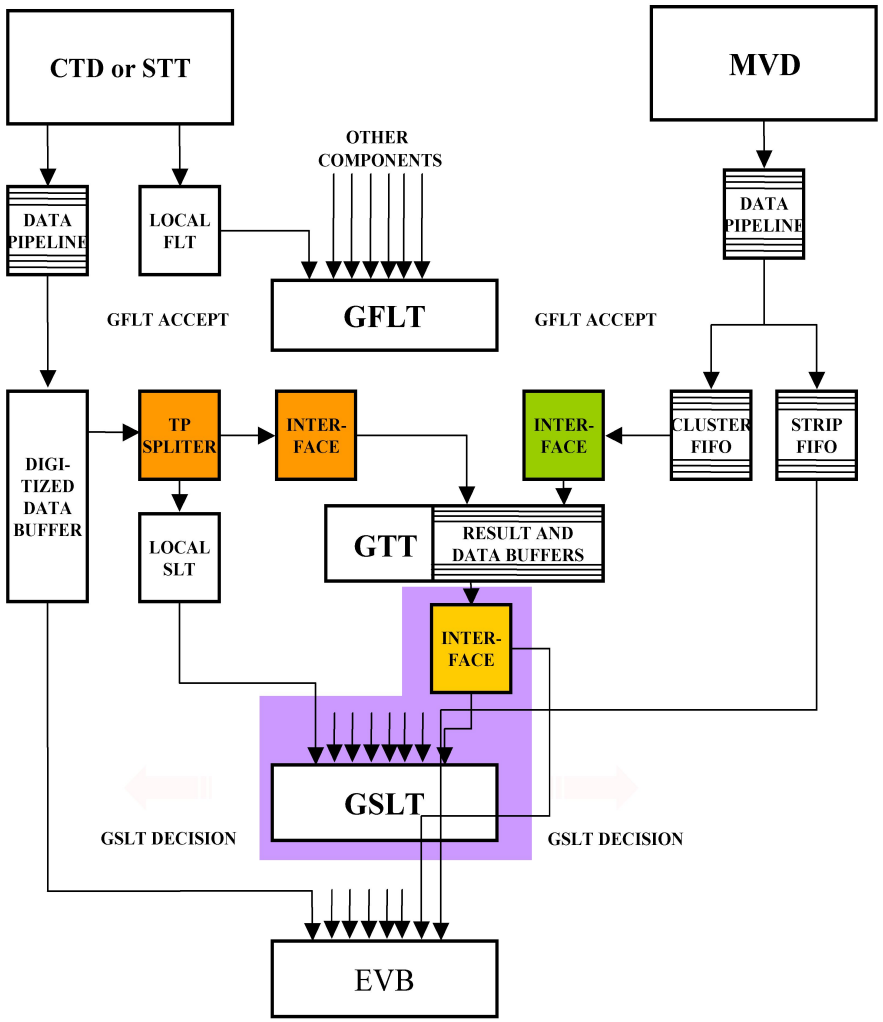
Overview of the GTT system

- Detector data pipelined at bunch crossing (10MHz). On GFLT accept..
- Transputer based readout of CTD and STT (∼3 ms). Within the Transputer system data is forked and sent to:
  - ▷ GTT component interface,
  - ▷ Existing component SLT. On a GSLT accept also to the Event Builder.
- MVD-GTT component interface integrated with MVD VME readout (∼1 ms).
- Component VME interfaces ship component raw data to GTT 1GHz Linux PC farm.
- Algorithm result banks sent to Global Second Level Trigger (GSLT).



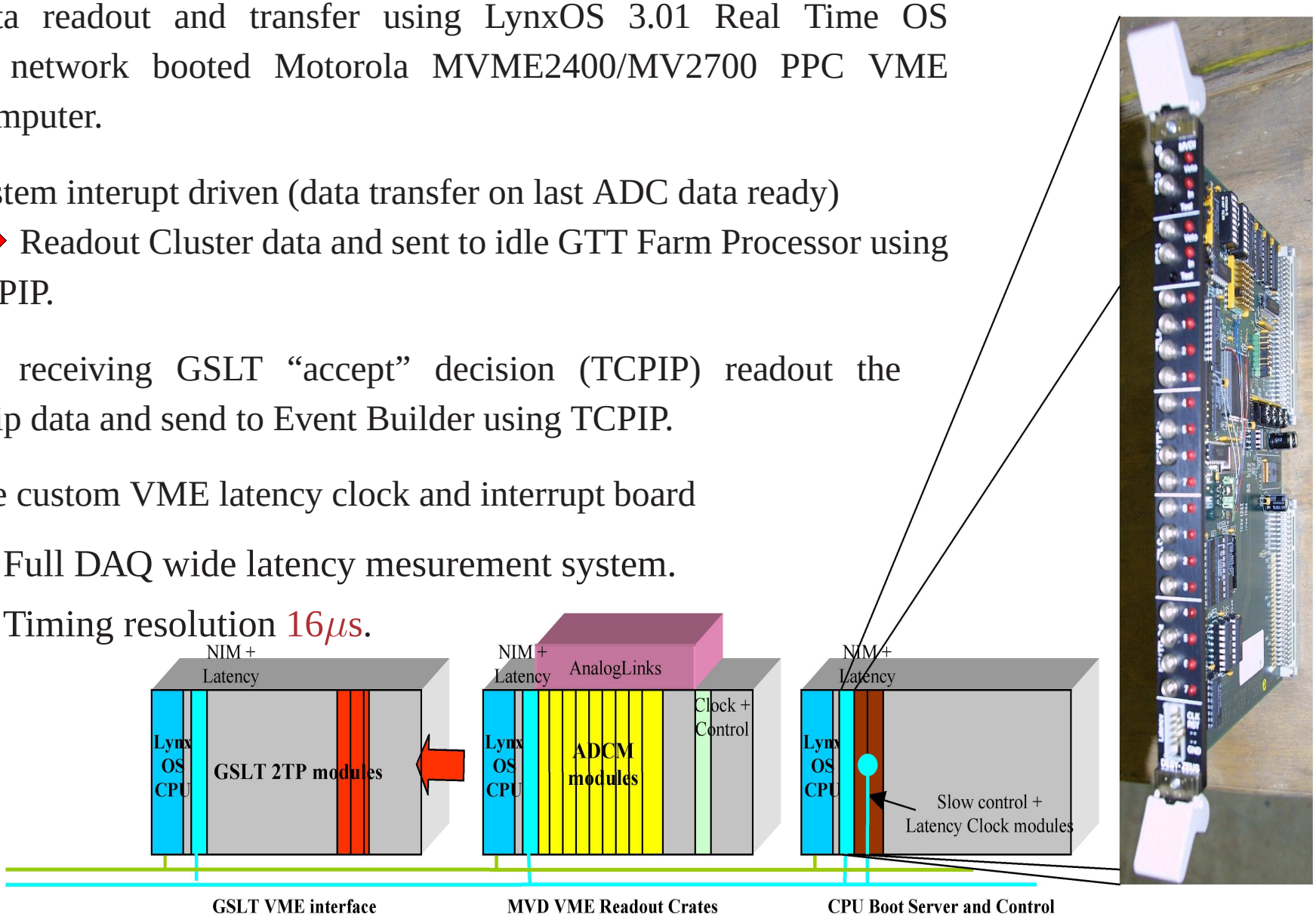
Overview of the GTT system

- Detector data pipelined at bunch crossing (10MHz). On GFLT accept..
- Transputer based readout of CTD and STT (∼3 ms). Within the Transputer system data is forked and sent to:
  - ▷ GTT component interface,
  - ▷ Existing component SLT. On a GSLT accept also to the Event Builder.
- MVD-GTT component interface integrated with MVD VME readout (∼1 ms).
- Component VME interfaces ship component raw data to GTT 1GHz Linux PC farm.
- Algorithm result banks sent to Global Second Level Trigger (GSLT).

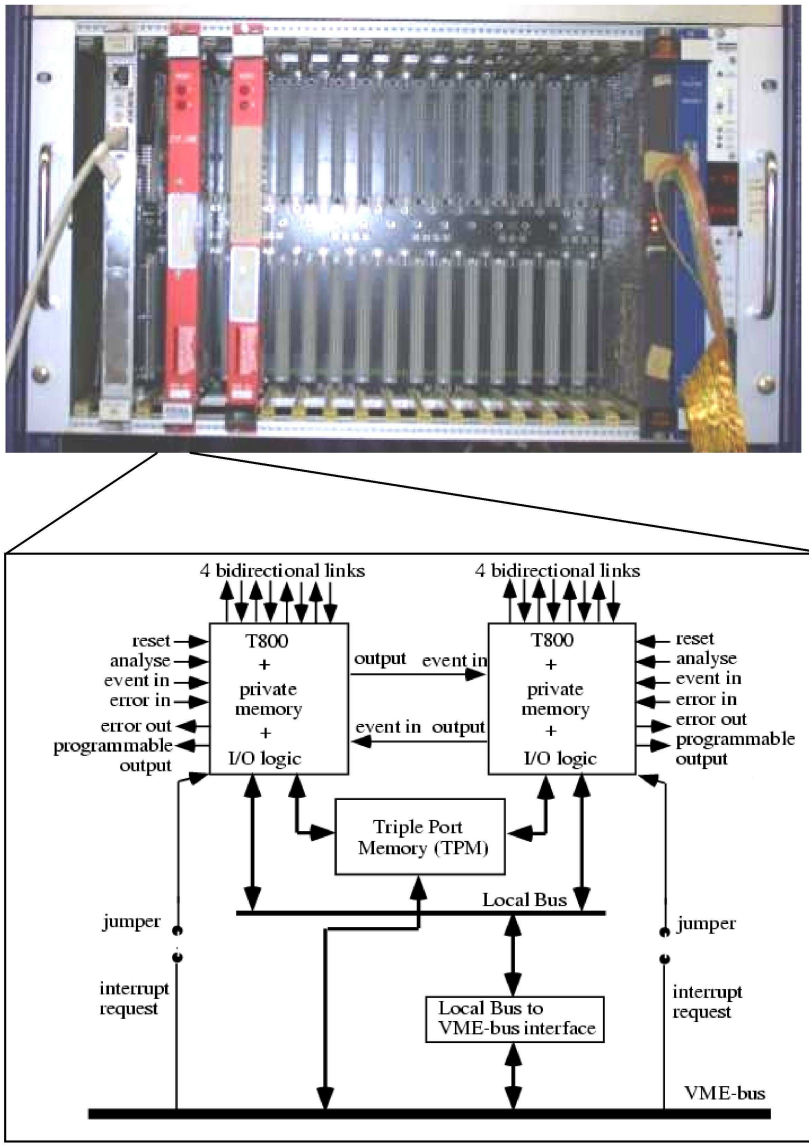


The MVD Component Interfaces

- Data readout and transfer using LynxOS 3.01 Real Time OS on network booted Motorola MVME2400/MV2700 PPC VME Computer.
- System interupt driven (data transfer on last ADC data ready)
  ⇒ Readout Cluster data and sent to idle GTT Farm Processor using TCPIP.
- On receiving GSLT "accept" decision (TCPIP) readout the Strip data and send to Event Builder using TCPIP.
- Use custom VME latency clock and interrupt board
  - ▷ Full DAQ wide latency mesurement system.
  - ▷ Timing resolution 16μs.



The CTD and STT Component Interfaces

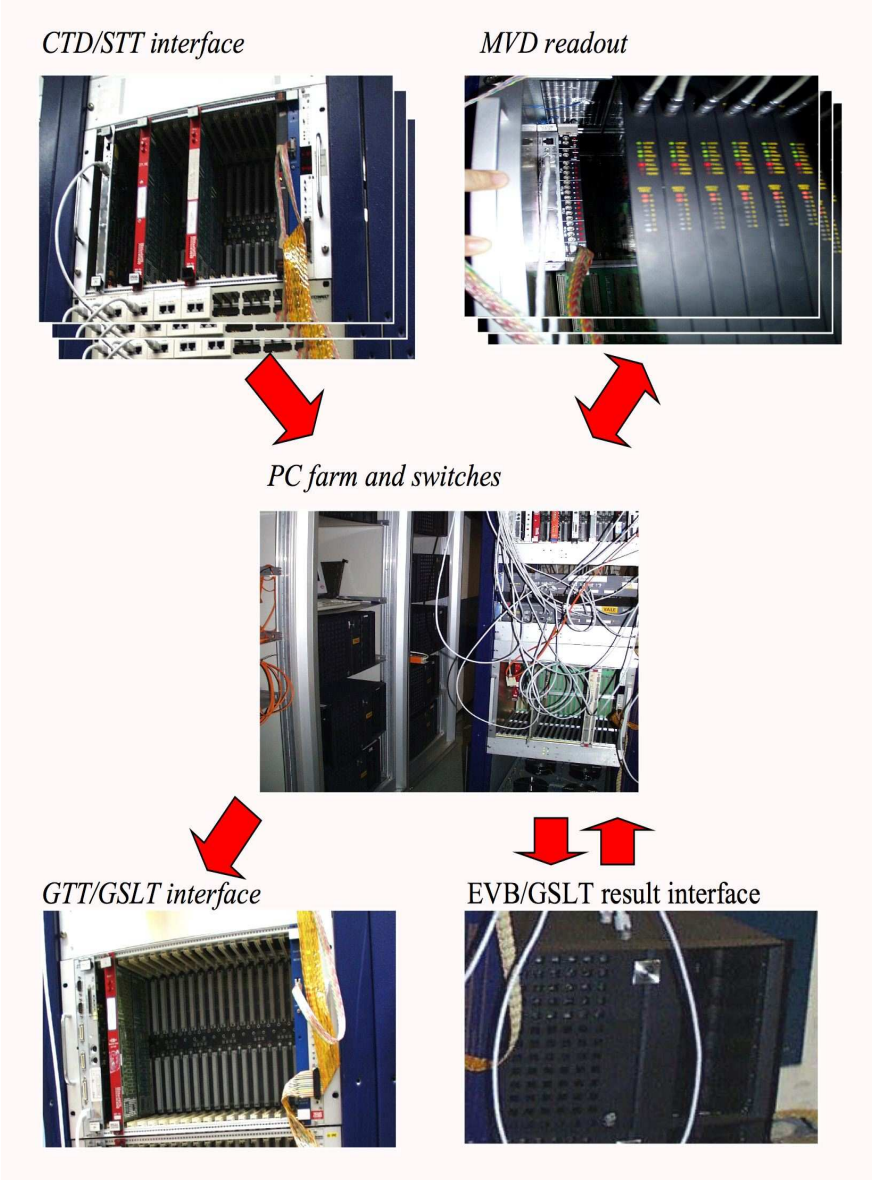


- Transputer based CTD and STT readout electronics.
- Trigger event data is split in Transputer networks and sent over TP links to NIKHEF 2TP[a] boards and copied to Triple Port Memory.
  - ▷ CTD, STT design mean data sizes ∼5 kB.
- Event data copied via VME backplane to LynxOS CPU.
- Data sent with FastEthernet TCP to GTT Farm Processor.

[a] Nikhef NIM **A332** (1993) 263

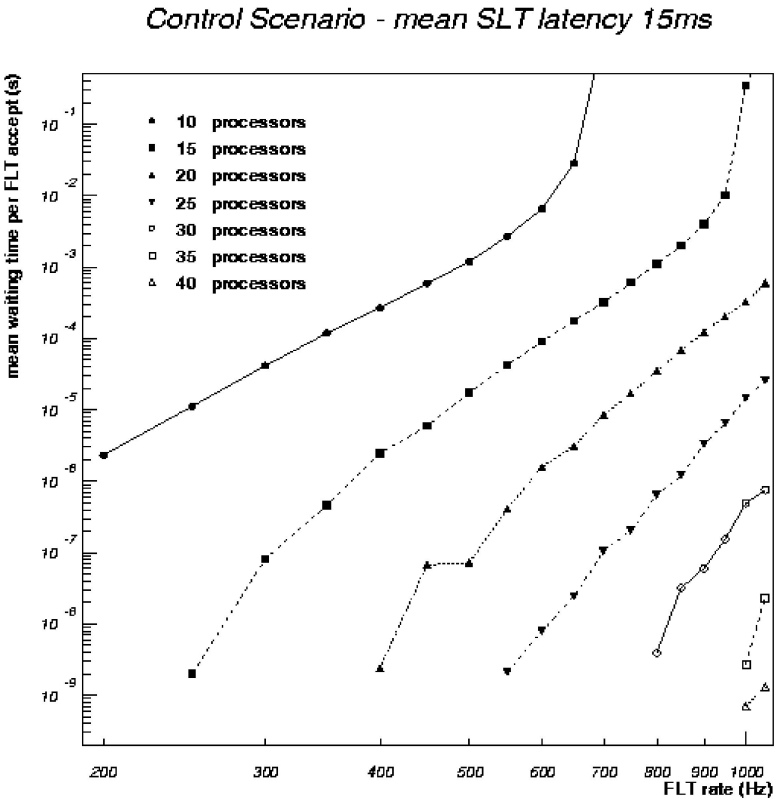
The GTT Hardware

- MVD readout:
  3 Motorola MVME2400 450MHz.
- CTD, STT interfaces:
  Motorola MVME2400 450MHz,
  2×NIKHEF-2TP VME-Transputer.
- PC Farm:
  12× Dell PowerEdge 4400 Dual 1GHz.
- GTT-GSLT interface:
  Motorola MVME2700 367MHz.
- GTT-EVB Trigger Decision interface:
  Dell PowerEdge 6450 Quad 700MHz
- Network switches:
  3× Intel Express 480T Fast/Giga 16 ports.
- Thanks to Intel Corp. who provided switches and PowerEdge hardware via Yale grant.



The GTT Processor Farm

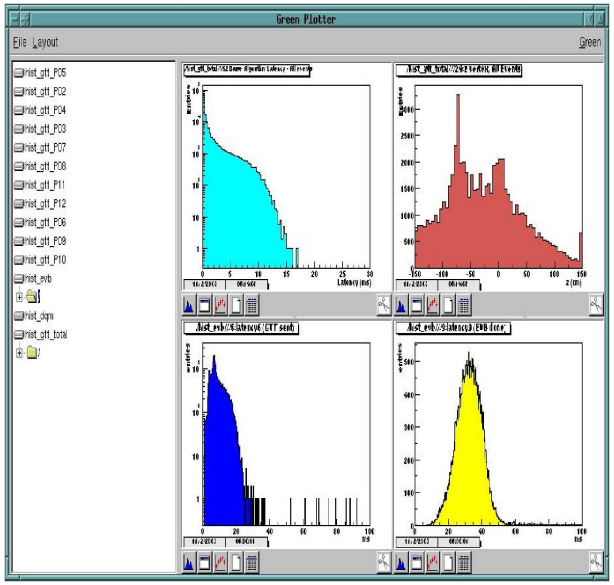
- Discrete event simulation including multiple data sources and processors.
  - ▷ Realistic GFLT arrival times.
  - ▷ Independent component readout latency.
  - ▷ Realistic algorithm processing latency.
  - ▷ Simulate holding the GFLT busy.
  - ▷ Ignore Data transfer latency and contention within the network switch.

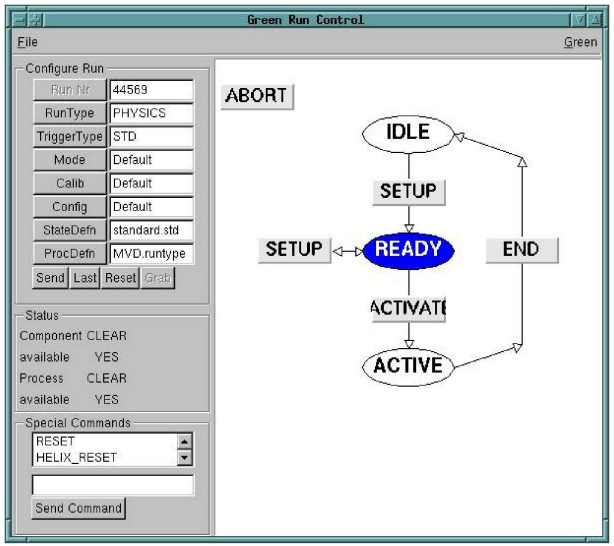


- Assume higher GFLT rate than expected.
- GTT latency must not be worse than existing CTD component.
- Exclude Round Robin scenario
- Control Credit based distribution based on first available GTT node ⇛ ∼10 nodes required.

GTT Node and Configuration Control

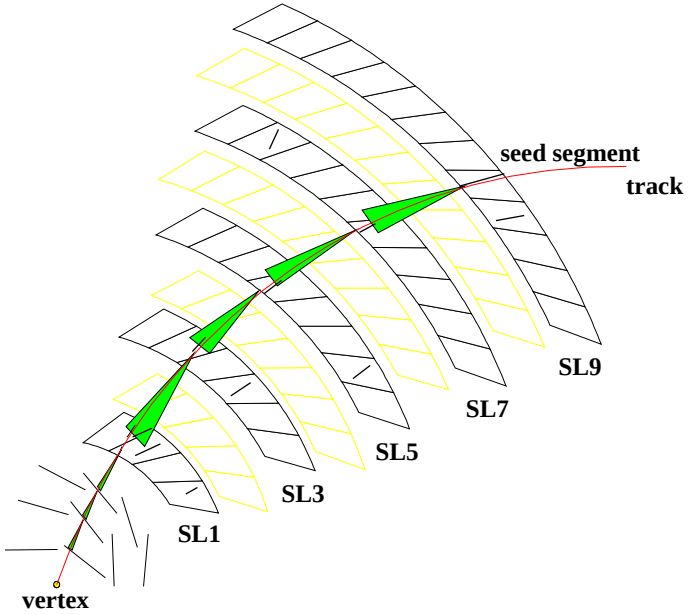
- Multi-threaded algorithm process:
  - ▷ 1 thread per input data source.
  - ▷ 1 thread per algorithm
    ➠ (Barrel=CTD+MVD, Forward=STT+MVD).
  - ▷ 1 timeout thread sending PASS after 40ms.
- Plot server pushing shared memory histograms.
- Statistics server pushing shared memory event statistics.
- "Playback" system, injecting Monte Carlo or Dumped data into component interfaces.
- State transition diagram and Local Run Control.
- Active nodes configured automatically on SETUP.
- Interprocess message passing using Fixed length (64 byte) XDR header (GLT number etc) and optional data block.

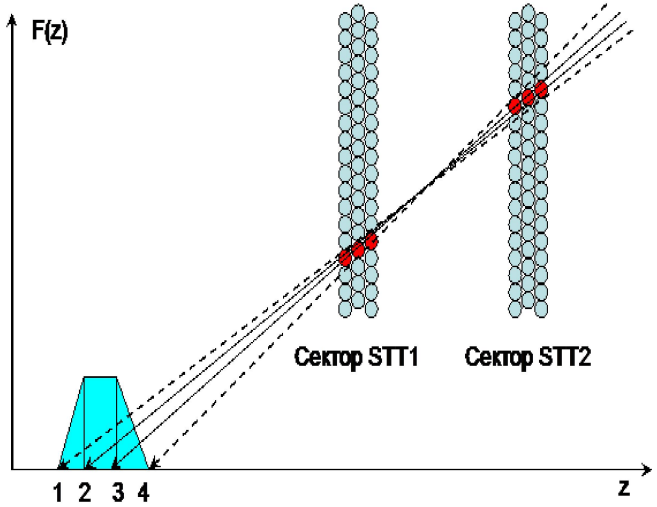




The Barrel and Forward Algorithms

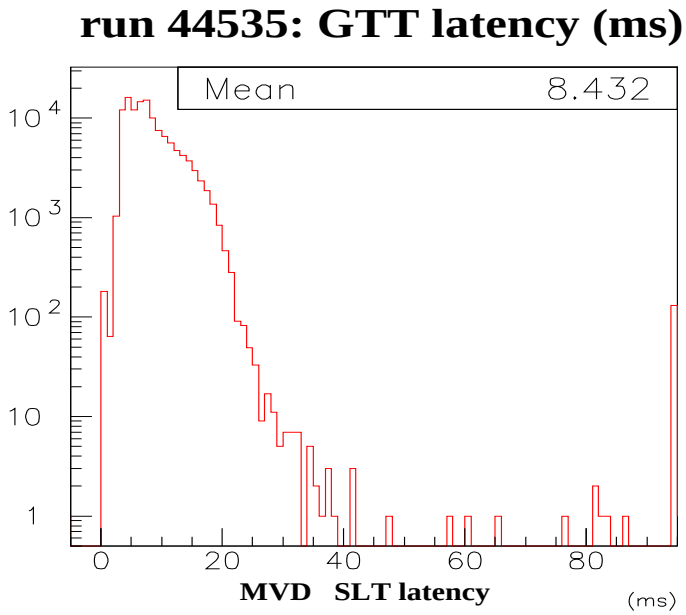
- Barrel Algorithm ...
  - ▷ Find tracks in the CTD, extrapolate to MVD to resolve pattern recognition ambiguity.
  - ▷ First find tracks in $r - \phi$.
  - ▷ Match z hits from CTD z-by-timing system and stereo layers.
  - ▷ Match MVD hits (currently disabled)
  - ▷ Fit vertex.

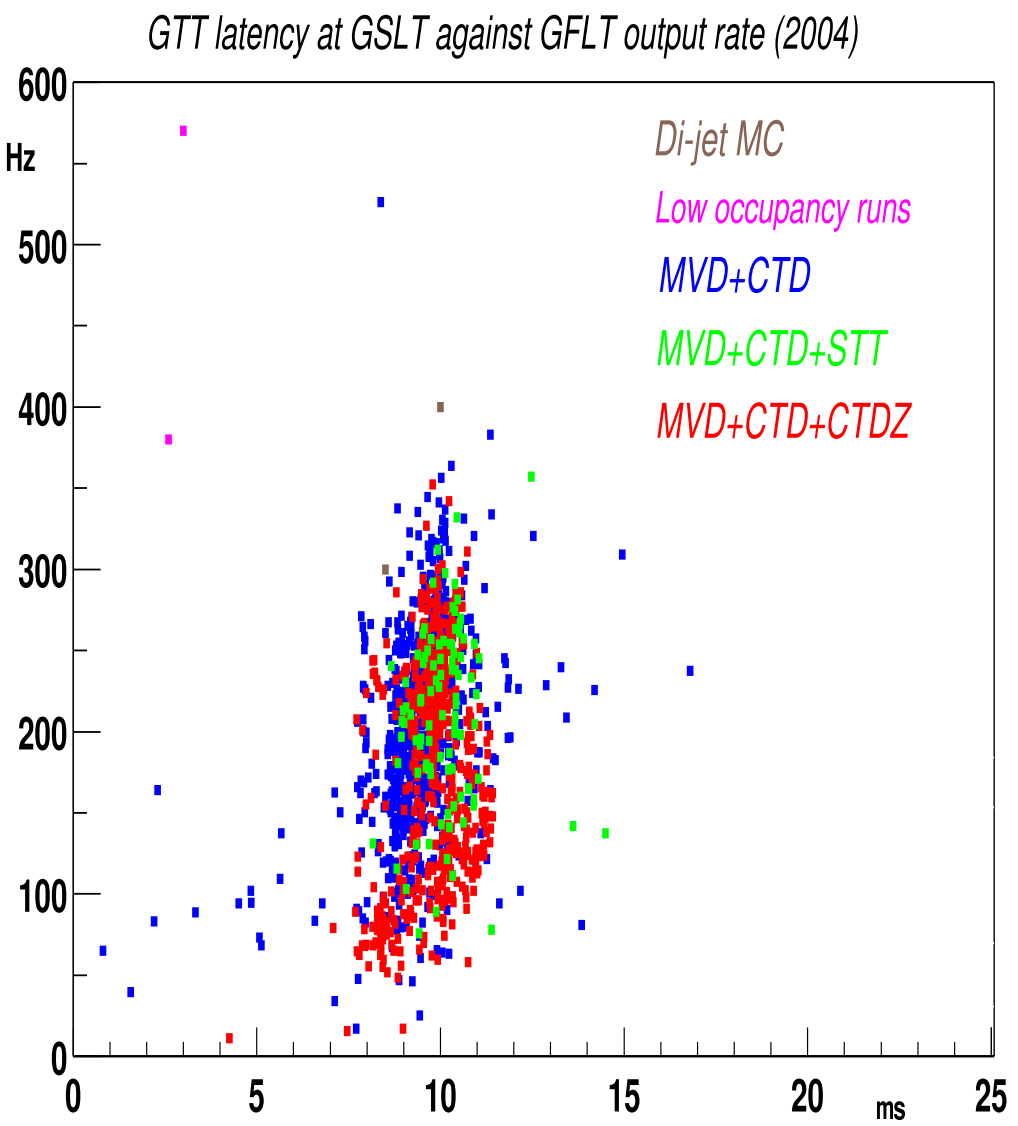




- Forward Algorithm (Not yet running as standard) ...
  - ▷ Match hit straws in parallel STT sectors with same ϕ and straw position.
  - ▷ Build segment projection envelope at beam line.
  - ▷ Fit vertex from all envelope projections.
  - ▷ Adding Forward MVD hits under development.

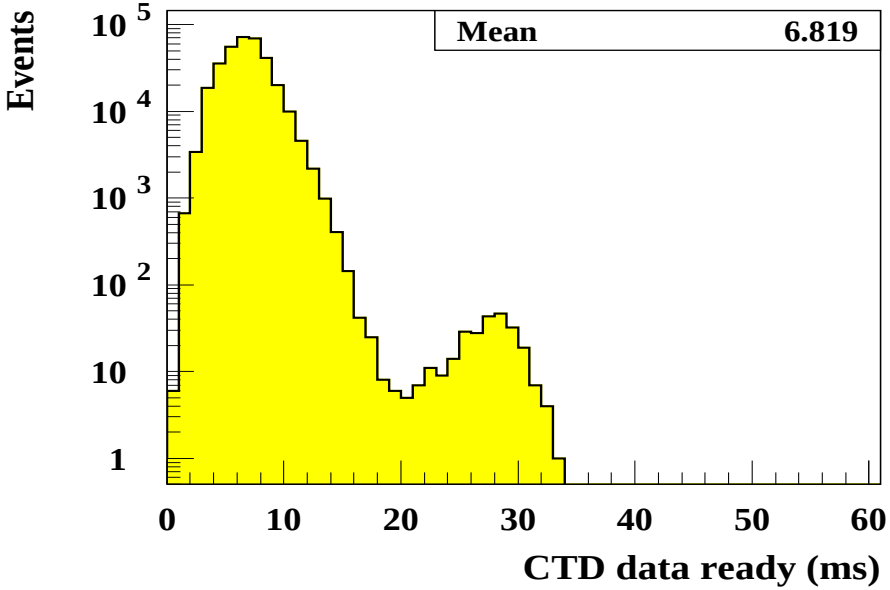
System Latency

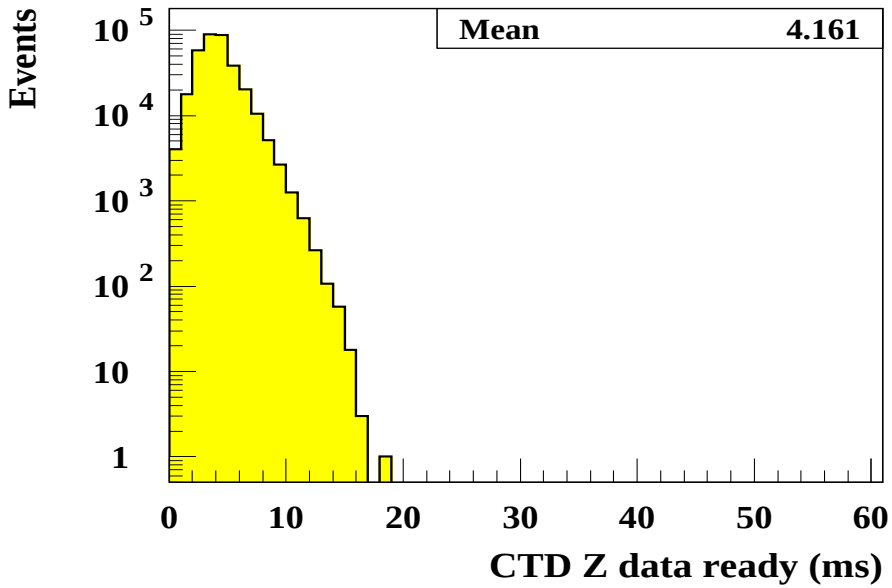


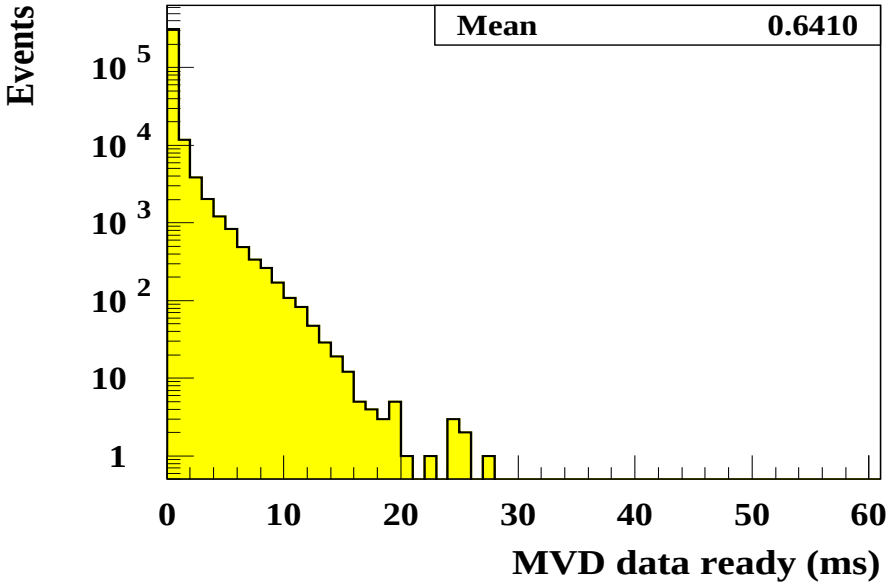


- Functioned smoothly throughout 2003-04 data taking ⇒ over 40 pb^{-1} on tape.
- Monte Carlo latency in playback mode consistent with data.
- Small dependence on GFLT rate ⇒ rates ∼500 Hz cause no additional deadtime.

Latency Breakdown - Data Ready

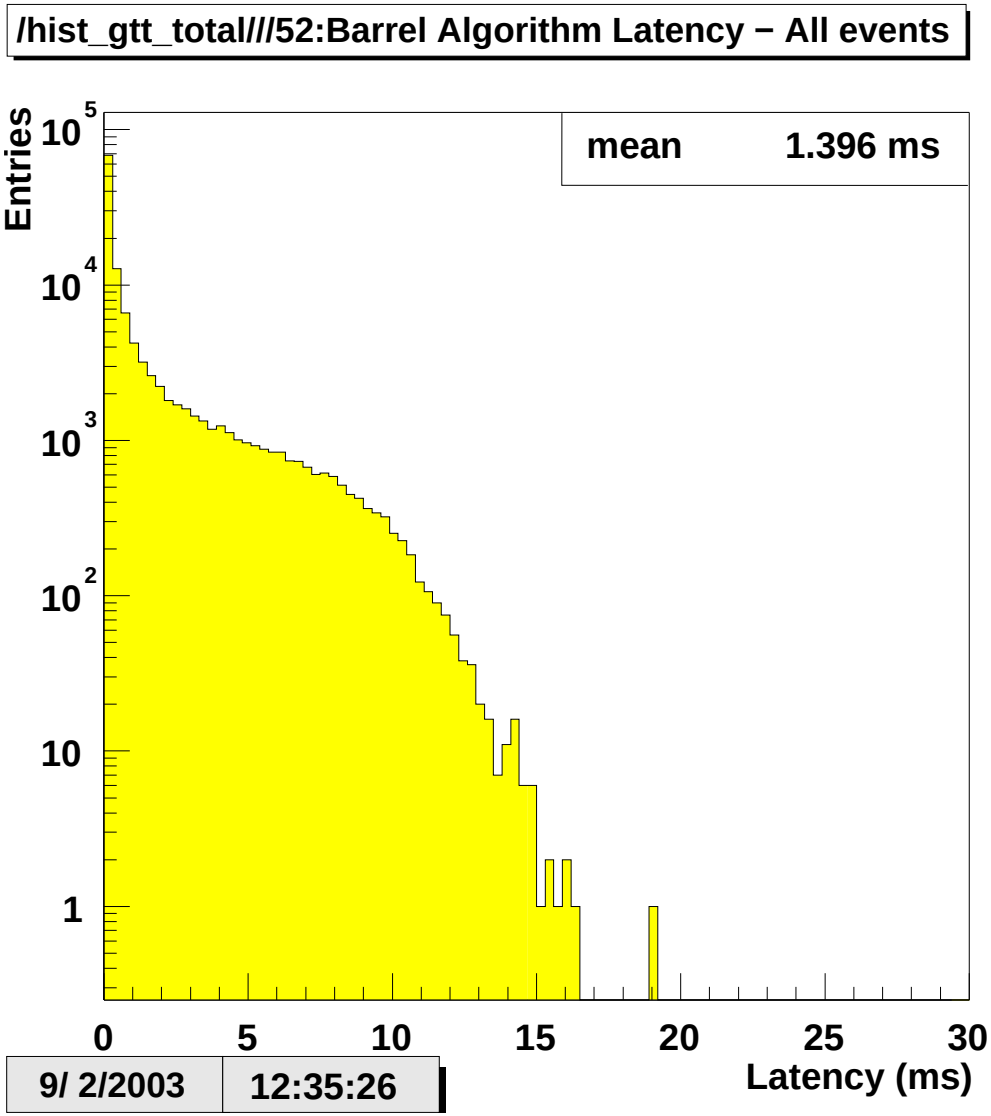


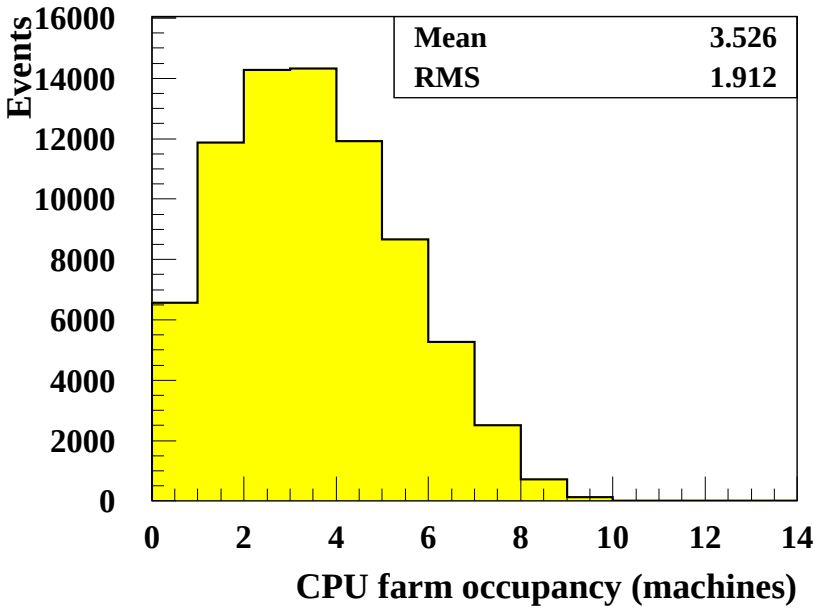




- Data transfer time and algorithm processing strongly correlated with data volumes.
- Mean data volumes after cut offs ...
  - ▷ MVD Cluster data – 0.4 kB
  - ▷ CTD axial and stereo data – 3.1 kB
  - ▷ CTD z-by-timing data – 0.8 kB
- Largest transfer latency CTD data $\sim$6.8 ms.

Latency Breakdown - Barrel Algorithm

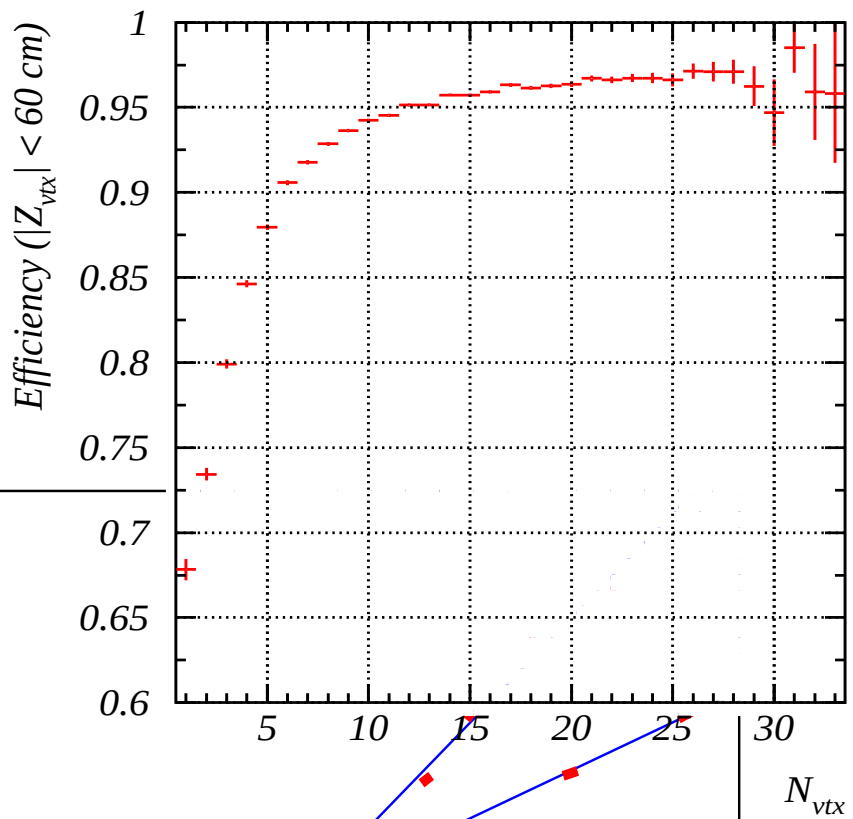


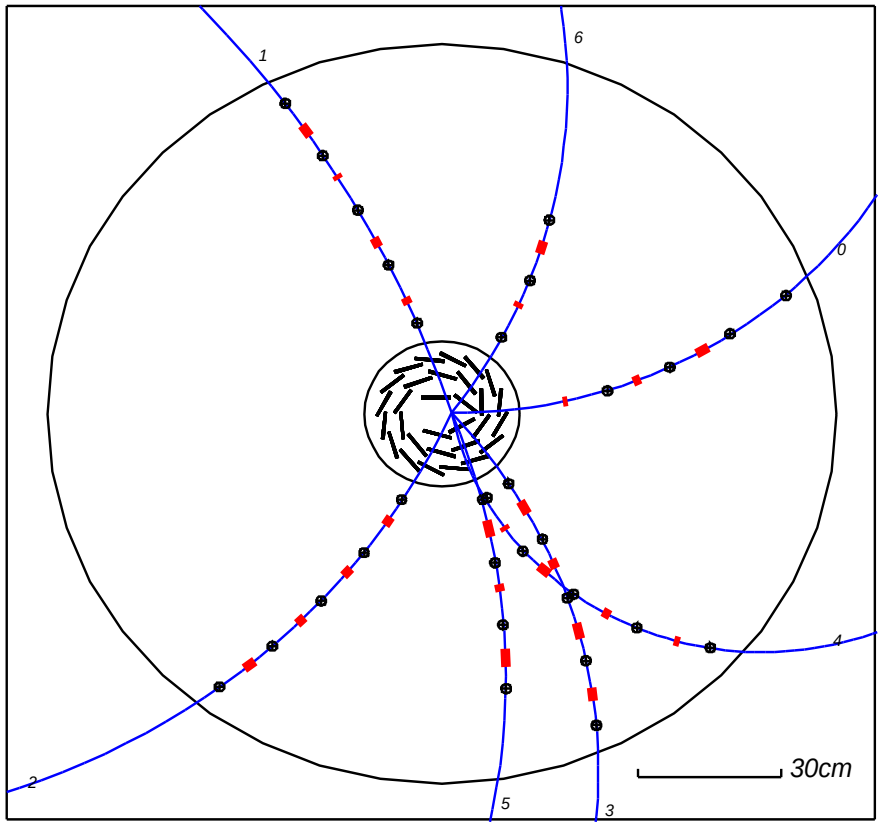


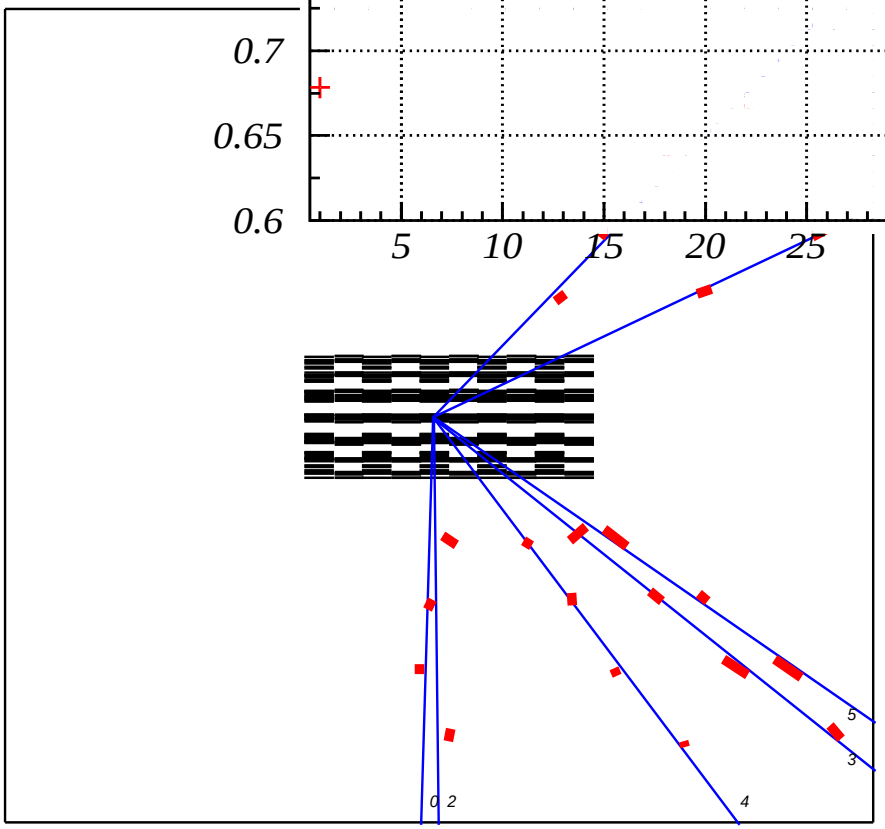
- Mean latency at GSLT, ~ 9 ms.
- Mean barrel algorithm latency ~ 1.4 ms
  ⇒ CPU Farm occupancy realatively low.
- Overall latency dominated by transfer time of large data volumes.

Performance – Barrel Algorithm

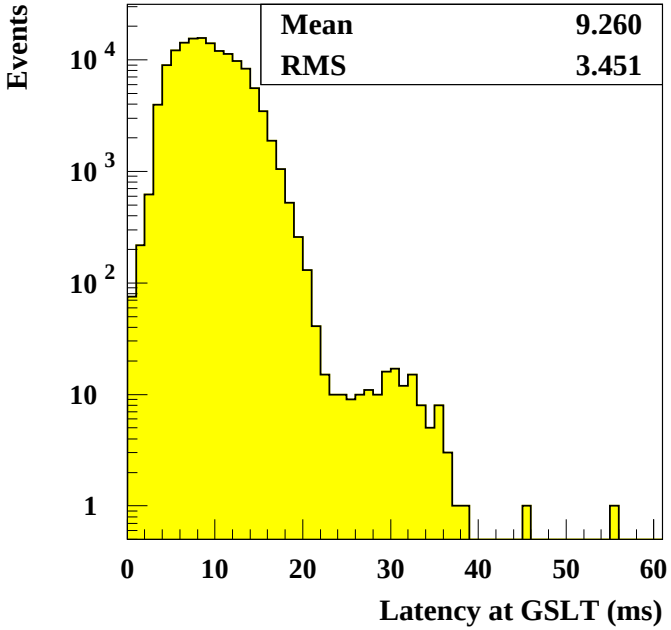
- Reconstructs complex event topologies.
- Vertex finding efficiency better than 90% for vertices with greater than 5 tracks on the vertex.

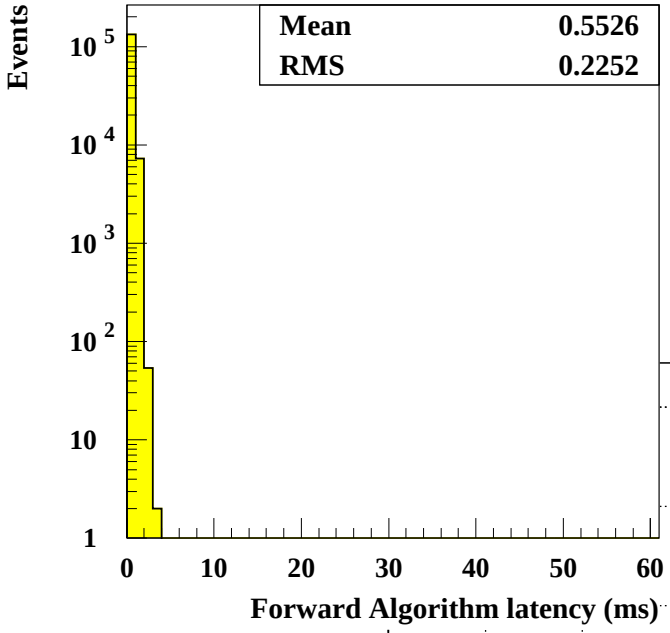




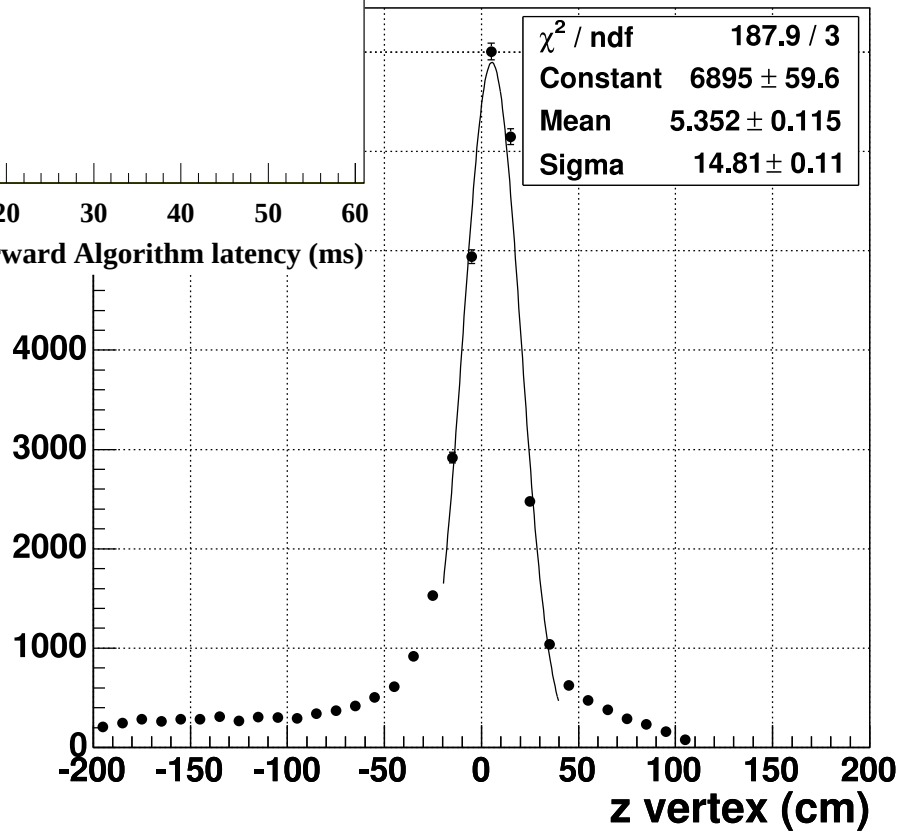


Forward Algorithm performance





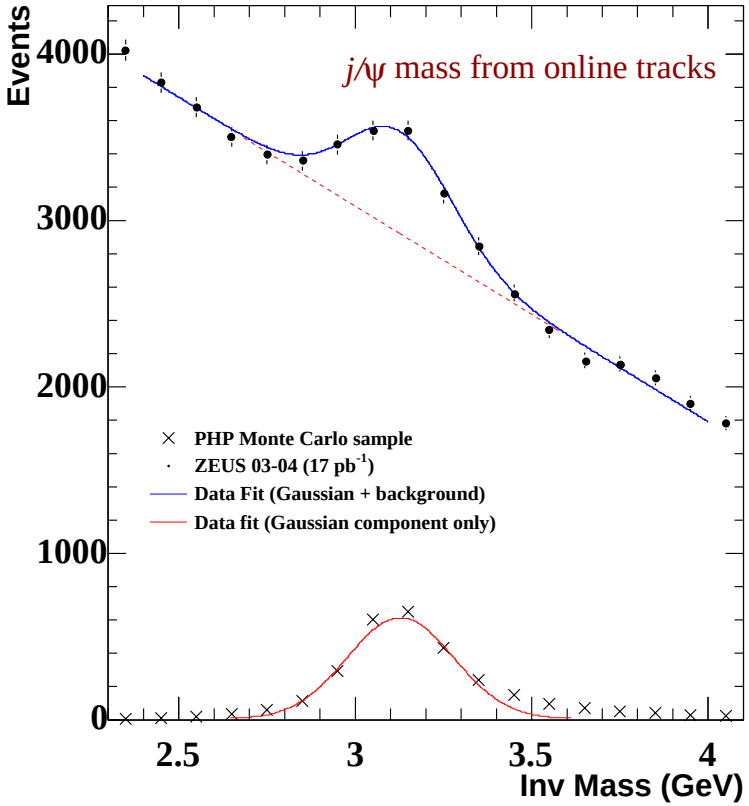
- A preliminary forward algorithm is available, and has been tested online.
- Larger than expected data volumes ⇒ tight data volume cuts.
- Algorithm runs stably ⇒ produces reasonable physics vertex.



Summary and Outlook

- GTT running: very stable and reliable throughout 2003-04 ⇒ over 40 pb^{-1} on tape.
- Fully integrated in ZEUS DAQ and Trigger system, selecting events online.

- For the HERA startup (October 2004)
  - ▷ Forward algorithm ready for parasitic running as standard ⇒ Forward tracking at the Second Level for the first time.
  - ▷ Include MVD hits in standard algorithm ⇒ greatly improve vertex resolution.



- Starting to address improved heavy flavour selection...
- Looking forward to more high luminosity data taking with improved Barrel and Forward Algorithms and event selection when HERA restarts in October.