A Software Data Transport Framework for Trigger Applications on Clusters
Requirements

- Alice: A Large Ion Collider Experiment
- Very large multiplicity: >15,000 particles/event
- Full event size > 70 MB
- Data rate into last trigger stage (High Level Trigger, HLT) up to 25 GB/s
- HLT has full event data available
Requirements

HLT primary task is event reconstruction for triggering and storage

From raw ADC values...

1, 2, 123, 255, 100, 30, 5, 1, 4, 3, 2, 3, 4, 5, 3, 4, 60, 130, 30, 5, ........

to

particle tracks
High Level Trigger Dataflow

- HLT consists of Linux PC farm w. roughly 1000 nodes
- Analysis is performed hierarchically
- Several stages for data processing and merging
- Natural mapping of dataflow, cluster topology, detector geometry
High Level Trigger

Framework software required to transport data in HLT

• Flexible
  - Components communicating via standardized interface
  - Pluggable components to support different configurations
  - Support for runtime reconfiguration

• Efficiency
  - Minimize CPU usage to retain cycles for data processing (primary)
  - Transport data as quickly as possible (secondary)

• C++ Implementation
Component Interface

- Only locally on a node
- Uses shared memory for data and named pipes for descriptors
- Multiple consumers attached to one producer (Publisher–Subscriber paradigm)
- Buffer management has to be done in data producer
Network Communication

- Uses class library
- Abstract call interface
- Classes optimized for
  - Small message transfers
  - Large data blocks
- Implementations for multiple network technologies/protocols possible
- Currently supported: TCP & SCI
Components

Framework contains components to configure dataflow

- To merge data streams belonging to one event

- To split and rejoin a data stream (e.g. for load balancing)
Components

Framework contains components to configure dataflow

- To transparently transport data over the network to other computers (Bridge)

- SubscriberBridgeHead has subscriber class for incoming data, PublisherBridgeHead uses publisher class to announce data

- Both use network classes for communication
Components

Templates for user specific components

- Read data from source and publish it (Data Source Template)

- Accept data, process it, publish results (Analysis Template)

- Accept data and process it, e.g. storing (Data Sink Template)
Fault Tolerance

Components to handle software or hardware faults

- Processing distributed for load balancing + redundancy
- Upon failure reschedule events and activate spare node

Data Source

Fault Tolerance
Real-World-Test

HL3
Event Stream Merger

HL2
2*Patch Merger

HL1
2*Tracker

HL0
Clusterfinder
ADC Unpacker

Chep03 – UCSD – March 24th–28th 2003
T. M. Steinbeck, V. Lindenstruth, H. Tilsner, for the Alice Collaboration
Task:

- Tracking of simulated Alice pp events,
- Pile up of 25 events
- Simulate one sector of Alice TPC
  (1/36 of detector)

Performance on mix of 800 MHz and 733 MHz systems:

Processing rate of more than 420 Hz
Fault Tolerance Test

Curves are scaled independently and arbitrarily

T. M. Steinbeck, V. Lindenstruth, H. Tilsner, for the Alice Collaboration
# Interface Performance

<table>
<thead>
<tr>
<th></th>
<th>733 MHz PC</th>
<th>800 MHz PC</th>
<th>933 MHz PC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average Event Rate [kHz]</strong></td>
<td>11.86</td>
<td>12.73</td>
<td>14.41</td>
</tr>
<tr>
<td><strong>Average Time Overhead [(\mu s/event)]</strong></td>
<td>168.7</td>
<td>157.1</td>
<td>138.8</td>
</tr>
</tbody>
</table>

![Graph showing scaling between 733 MHz, 800 MHz, and 933 MHz](image)
Interface Performance

Reference PC memory benchmark scaling

Scaling 733 MHz - 800 MHz - 933 MHz

Wed Mar 19 19:56:27 2003

T. M. Steinbeck, V. Lindenstruth, H. Tilsner, for the Alice Collaboration
Conclusion

- Working framework
- Flexible configuration w. fault tolerance abilities
- Can already be used in real applications

To Do:
- Tool for easier configuration and setup
- Fault tolerance control instance/decision unit
- Fine-grained fault tolerance and recovery
- More tuning