Fault Tolerance/Adaptation in the BTeV Trigger

- What is BTeV?
- The BTeV Trigger
- Fault Tolerant and Fault Adaptive Strategies
- Conclusions

CHEP'04
INTERLAKEN

Paul Sheldon
Vanderbilt University

The Next Generation

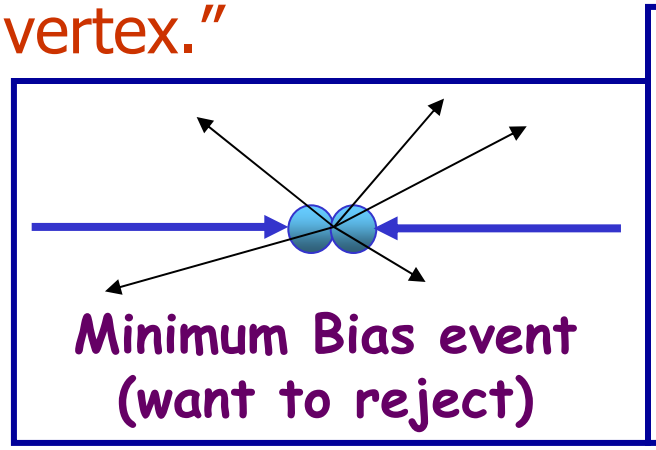
- A 2nd generation of B-factories at hadron machines: BTeV and LHC-b
  - both will run in the LHC era.

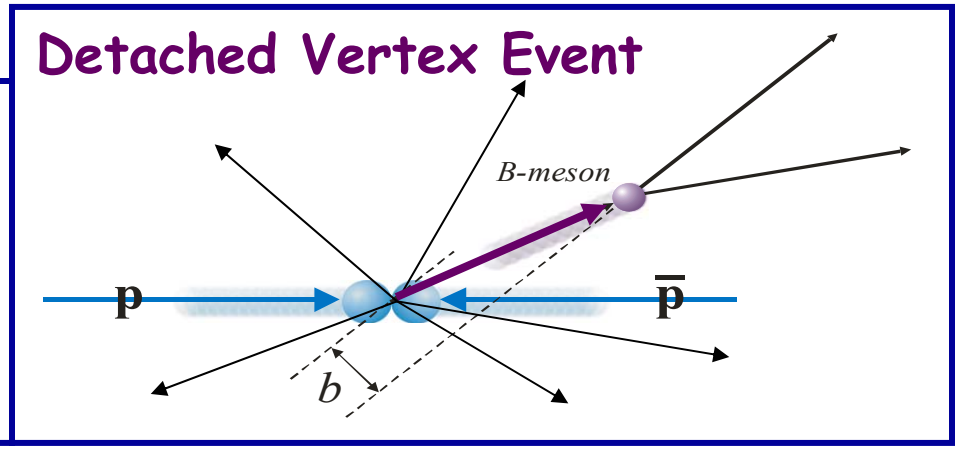
- Why at hadron machines?
  - ~**10^{11}** b hadrons produced per year (10^7 secs) at 10^{32} $cm^{-2}s^{-1}$
  - **e^+e^-** at $\Upsilon(4s)$: ~**10^8** b produced per year (10^7 secs) at 10^{34} $cm^{-2}s^{-1}$
  - Get all varieties of b hadrons produced: B_s, baryons, etc.
  - Charm rates are 10x larger than b rates…

- Hadron environment is challenging…
  - CDF and D0 are showing the way
- To do the reqd. physics a dedicated experiment is needed
- BTeV: trigger on detached vertices at the first trigger level
  - Preserves **widest possible spectrum** of physics – **a requirement**.

The Challenge

- We want to select events (actually crossings) with evidence for a "downstream decay", a.k.a "detached vertex."





- We want to do this from the beginning, to maximize our efficiency for the widest spectrum of physics.
- This requires sophisticated track and vertex reconstruction in the first level of the trigger.
- Commodity computing in the first level (and all levels) of our trigger.

...The Challenge

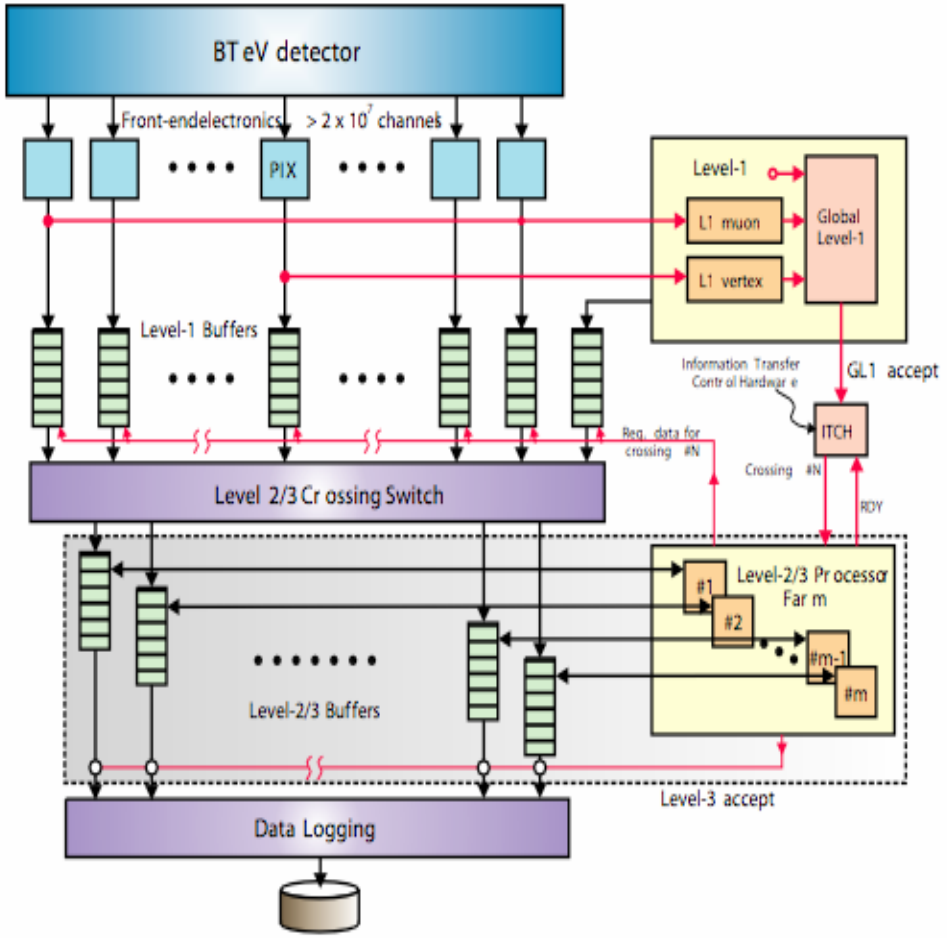
- The highly variable complexity of different crossings means that there will be a wide spread in event processing times.

- To have an efficient system, we must avoid idle time. This means pipelining all calculations and :
  - No fixed latency
  - No requirement of time ordering

- In turn, this implies
  - Massive amounts of buffering (TByte), and
  - On the fly sparsification in the front ends at the beam crossing rate

A 20 TeraHz Real-Time System

- **Input:** 800 GB/s (2.5 MHz)
- **Level 1:**
  reconstruct. takes 190μs,
  crossing rate of 396 ns:
  528 "8 GHz" G5 CPUs
  (factor of 50 reduction)
  high performance interconnects
- **Level 2/3:**
  Lvl 2 recon takes 5 ms
  (factor of 10 reduction)
  Lvl 3 recon takes 135 ms
  (factor of 2 reduction)
  1536 "12 GHz" CPUs
  commodity networking



- **Output:** 4 KHz, 200 MB/s
  (1—2 Petabytes/year)

The Problem

- Monitoring, Fault Tolerance and Fault Mitigation are crucial
  - In a large cluster of this size, processes and deamons are constantly hanging/failing without warning or notice
- Software reliability depends on
  - Detector-machine performance
  - Program test procedures, implementation, and design quality
  - Behavior of the electronics (front-end and within trigger)
- Hardware failures will occur: one to a few per week

Given the very complex nature of this system where thousands of events are simultaneously and asynchronously cooking, issues of data integrity, robustness, and monitoring are critically important and have the capacity to cripple a design if not dealt with at the outset… BTeV [needs to] supply the necessary level of "**self-awareness**" in the trigger system. [June 2000 Review]

BTeV's Response: RTES

- The **R**eal **T**ime **E**mbedded **S**ystem Group
- A collab. of five institutions, NSF ITR grant ACI-0121658
  - University of Illinois
  - University of Pittsburgh
  - University of Syracuse
  - Vanderbilt University (PI)
  - Fermilab
- Physicists and Computer Scientists/Electrical Engineers at BTeV institutions with expertise in
  - High performance, real-time system software and hardware,
  - Reliability and fault tolerance,
  - System specification, generation, and modeling tools.
- A group working on **fault management in large computing clusters** with the BTeV Trigger as its **testbed**

BTeV RTES Goals

- **High availability:** a **fault handling infrastructure** capable of
  - Accurately identifying problems (where, what, and why)
  - Compensating for problems (shift the load*, changing thresholds*)
  - Automated recovery procedures (restart / reconfiguration)
  - Accurate accounting
  - Being extended (capturing new detection/recovery procedures)
  - Policy driven monitoring and control
- **Dynamic reconfiguration:** adjust to potentially changing resources.
- Faults must be corrected in the shortest possible time, and corrected **semi-autonomously** (i.e. with as little human intervention as possible). Hence **distributed** and **hierarchical monitoring** and **control** are vital.
- **Life-cycle maintainability** and **evolvability** to deal with new trigger algorithms, new hardware and new versions of the OS

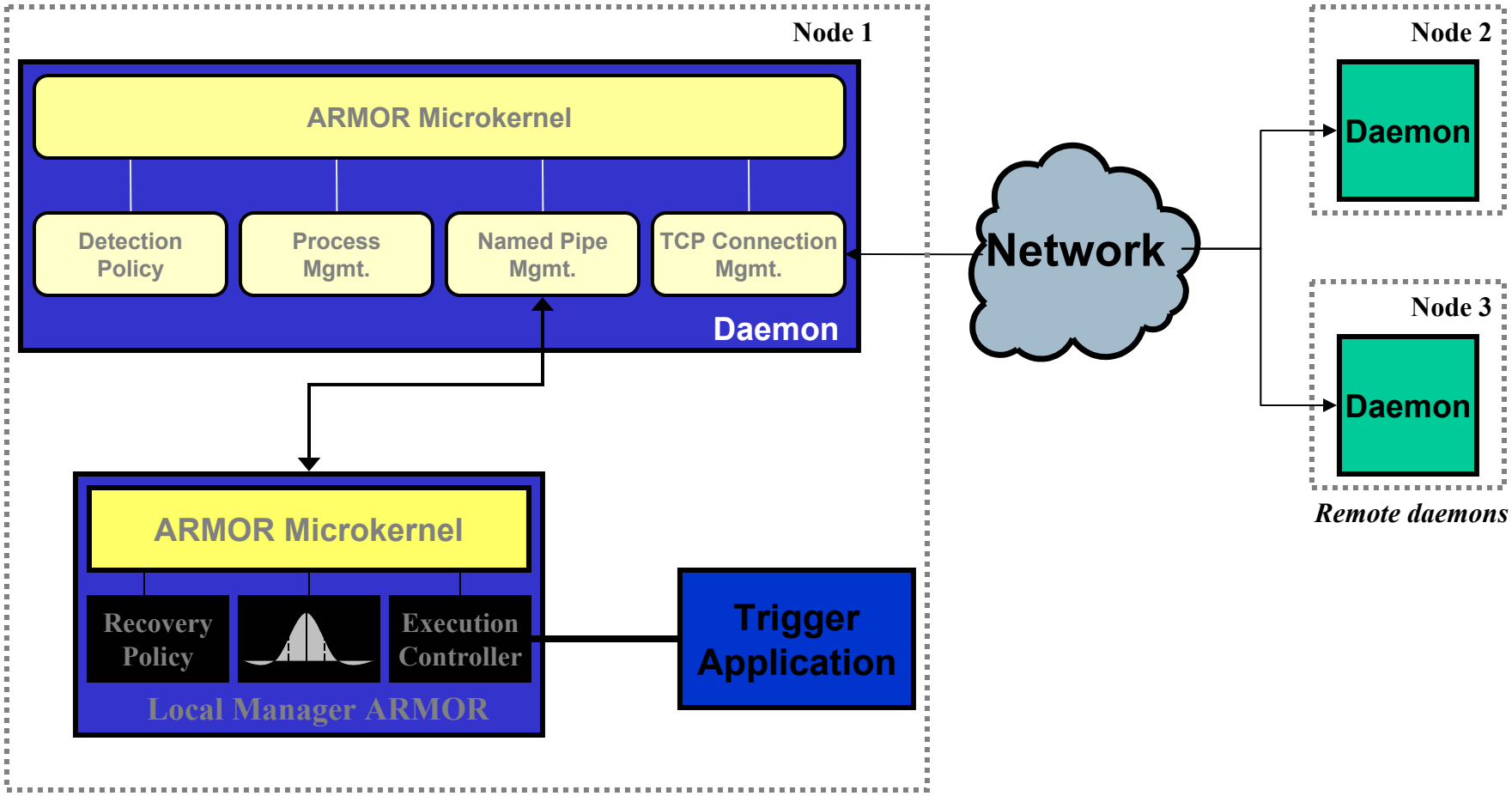
RTES Deliverables

- A hierarchical fault management system and toolkit:
  - **VLA**s (Very Lightweight Agents for limited resource environments)
  - **ARMOR**s (Adaptive, Reconfigurable, and Mobile Objects for Reliability – a framework for detection and reaction to faults)
  - **GME** (Generic Modeling Environment) system modeling tools and a BTeV specific **"**language**"** for modeling fault behaviors and trigger configuration and handling
- BTeV trigger and DAQ specific "plug-ins" using the above toolkit, applied to both hardware and software

ARMORs

- Multithreaded processes composed of replaceable building blocks called Elements
- Provide error detection and recovery services to the trigger and other applications
  - Restarts, reconfiguration
  - Removal from service
- A Hierarchy of ARMOR processes form a reconfigurable runtime environment:
  - System management, error detection, and error recovery services are distributed across ARMOR processes
  - ARMORE elements can monitor data quality, data rates,…
  - ARMOR runtime environment *can handle self failure*
- ARMOR support for an application:
  - Completely transparent and external support
  - Instrumentation with ARMOR API

ARMOR View



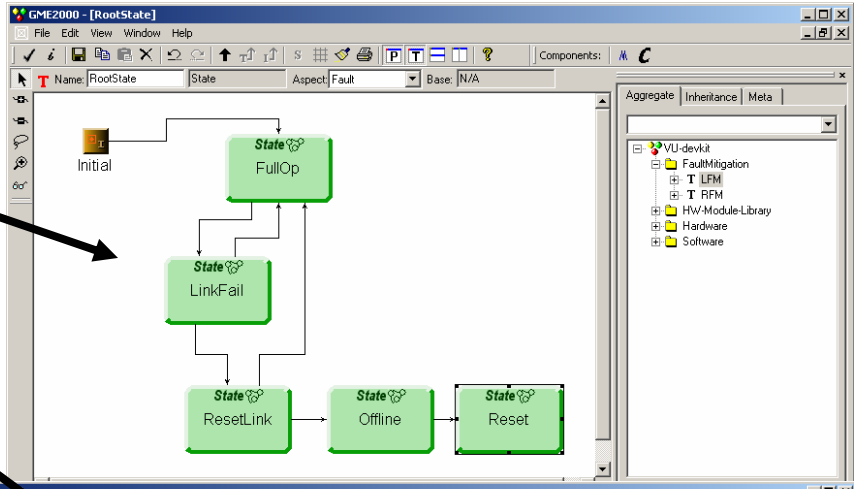
Adaptive, Reconfigurable, and Mobile Objects for Reliability

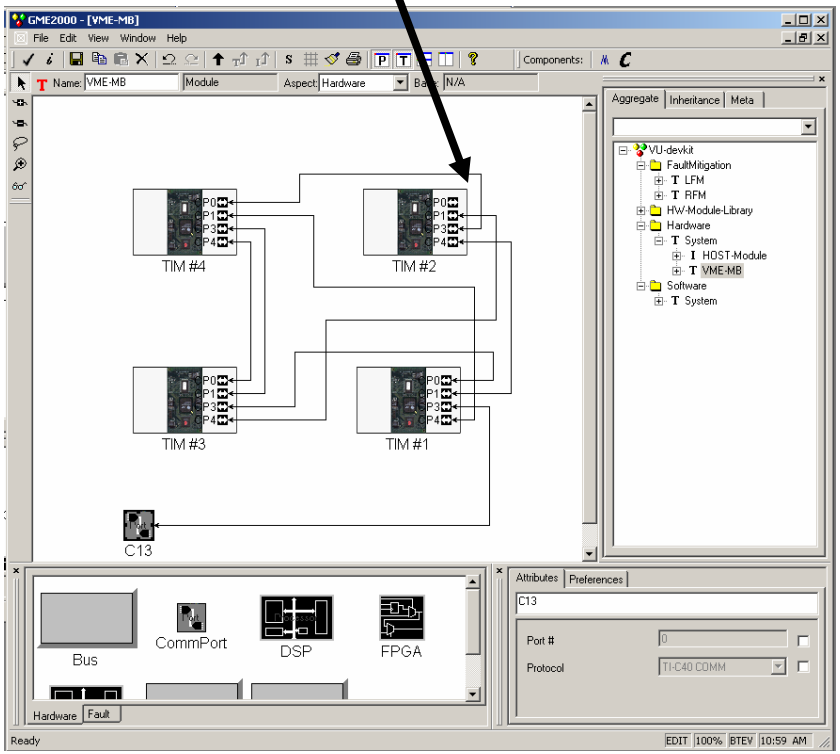
Configuration through Modeling

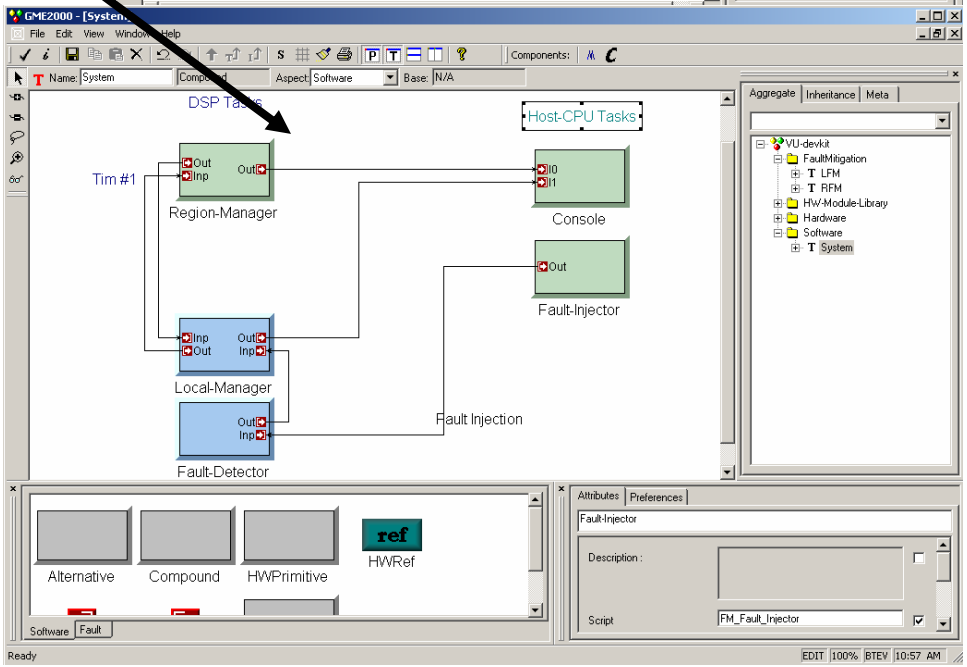
- Multi-aspect tool, separate views of
  - Hardware – components and physical connectivity
  - Executables – configuration and logical connectivity
  - Fault handling behavior using hierarchical state machines
- Model interpreters can generate the system image
  - At the code fragment level (for fault handling)
  - Download scripts and configuration (event filter)
- Modeling "language" is BTeV specific
  - Shapes, properties, associations, constraints
- First attempts at capturing physics run types, configuration information, and fault handling

Modeling Environment

Fault handling
Process dataflow
Hardware Configuration







RTES Demos

- We are using an iterative approach to develop the toolkit through a series of projects of increasing complexity.
- Demo 1: Demonstrated at Super Computing 2003.
  - First integration of tools, ran on 4 processor Level 1 Trigger Hrdw
- Demo 2: To be demonstrated at IEEE RTAS 2005 (March)
  - **Demonstrate scaling behavior of tools** (36 processor cluster)
  - Demonstrate the ability of GME to accurately model components of the trigger and predict overall system behavior
  - Demonstrate ability to detect trigger application failure & capture the event data for use in the nightly build / unit test suite.
  - Show adoption of a coherent build system that allows for library and system releases and nightly unit and integrated testing
- Demo 3: Version 0 of code on 1/8 system (highway)
- Demo 4: Delivered code on 1/8 system

Comments

- This is an *integrated approach* – from hardware to physics algorithms
  - Standardization of resource monitoring, management, error reporting, and integration of recovery procedures can make operating the system more efficient and make it possible to comprehend and extend.
- There are real-time constraints that must be met
  - Scheduling and deadlines
  - Numerous detection and recovery actions
- The product of this research will
  - Automatically handle simple problems that occur frequently
  - *Be as smart as the detection/recovery modules plugged into it*
- The product can lead to better or increased
  - Trigger uptime by compensating for problems or predicting them instead of pausing or stopping a run
  - Resource utilization - the trigger will use resources that it needs
  - Understanding of the operating characteristics of the software
  - Ability to debug and diagnose difficult problems

Further Information

- General information about RTES
  - http://www-btev.fnal.gov/public/hep/detector/rtes/
- General information about BTeV
  - http://www-btev.fnal.gov/
- Information about GME and the Vanderbilt ISIS group
  - http://www.isis.vanderbilt.edu/
- Information about ARMOR technology
  - http://www.crhc.uiuc.edu/DEPEND/projects-ARMORs.htm
- Talks from our last workshop
  - http://false2002.vanderbilt.edu/program.php