

# L2 Node Status & Timeline

Kristian Hahn  
University of Pennsylvania  
06/11/04

# Big Picture

- Phase 1

- FrontEnd → SlinkMerger → PC (1) → L2TS → TS
- TLD monitoring & control

- for the node, this involves ...

- integrating the PC into the I/O chain
- further development of node software/hardware

# I/O Integration (then)

- PULSAR → PC(algo) → PULSAR
  - Limitations:
    - programmed VME L1A's
    - serial event processing
    - no L2TS communication

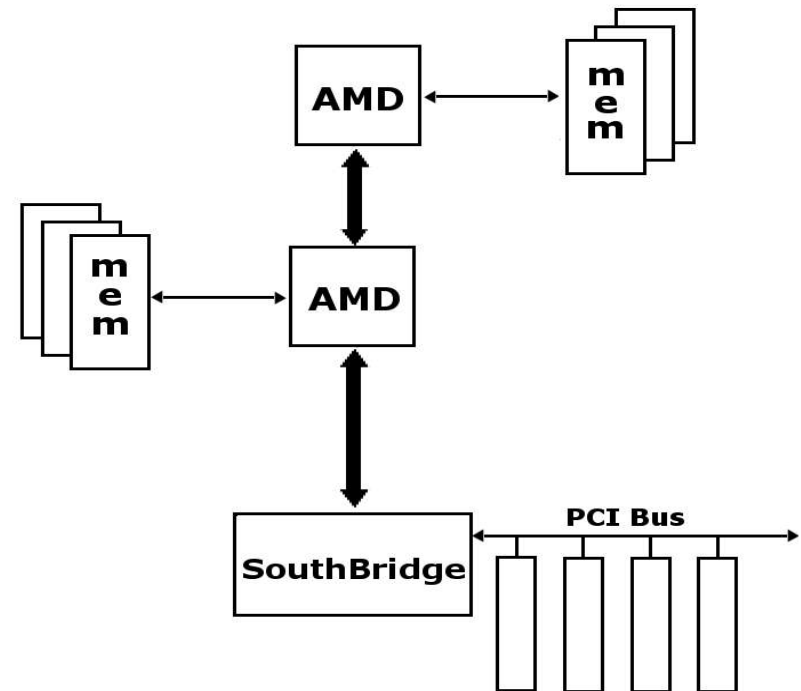
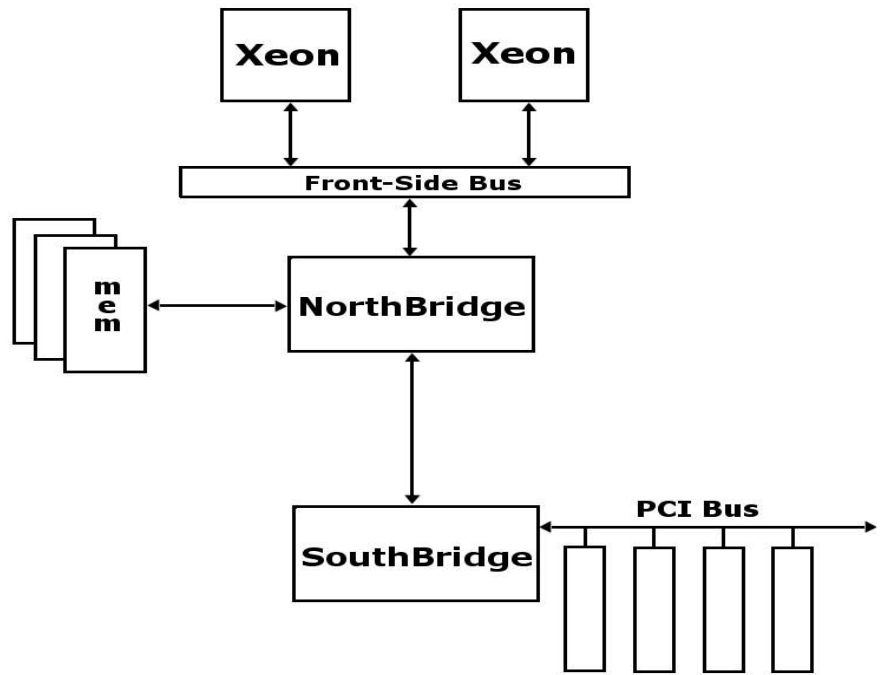
# I/O Integration (now & soon)

- Node I/O integration in stages ...
  - Slink-Tx(1) → PC(algo) → L2TS → TS
    - Input: load Slink-Tx with data of ~final format
    - Output: node generates formatted decisions
      - **Code tested with CJL, L2TS setup @ ~1KHz L1A rate**
      - **No TL2D generation**
  - Slink-Tx + SVT Tx/Rx → PC(algo) → L2TS → TS
    - Algos w/ 4 buffers and delayed SVT data on a separate link
      - **Now: Receive with delay and send canned decision to L2TS**
      - **Soon: Algo code for multiple buffers, not yet tested**
  - Slink-Tx + SVT Tx/Rx → Merger → PC(algo) → L2TS → TS
    - Input: PC configuration & data format as in Phase I
    - Output: TL2D sent with L2 accepts

# Node Development (hardware)

- New PC and CPU's
  - Dual 3.2GHz Intel Xeon upgrade
    - default for start of Phase I
  - Dual AMD Opteron 250/2.4GHz server
    - Alternate architecture, possible performance gain
      - **32/64 bit operation**
      - **Integrated memory controllers at processor speed**
      - **NUMA configuration, fast local memory access**
        - **Fully utilized in 64 bit OS's only** → **Gentoo Linux 2.6**
    - Develop alongside a *working* Xeon Phase I system
- S32PCI64 I/O
  - Switched from ODIN to HOLA
  - CERN drivers ported to and tested under Linux 2.6

# Node Development (architecture)



# Node Development (software)

- Control/Monitoring Interfaces (TLD)
  - Working model tested with Daniel
  - Config & monitoring methods still developing
- Upstream/Downstream I/O
  - Minimal changes to existing code
  - TL2D generation
- Algorithm Code
  - Prescaling, first pass done
  - Implement minimal triggers for running on first real data
  - Investigate trigger table switching mechanisms
  - *Gradually* rewrite the “Object Model”
    - Clean and easily maintainable C++
    - Bitfields vs. manual unpacking, saves time?

# Timeline

- June, week 3 :
  - Finish most I/O integration tests
  - Incorporate TLD monitoring/control in the tests
  - Code updates (data format, TL2D, monitoring/control )
- June week 4 :
  - Setup new hardware, move to trigger room
  - Complete node-side monitoring/control
  - Merger testing with full TLD control?
- July, week 1 :
  - Code cleanup, minimal triggers written
  - Runs with Merger and all available data paths?

# Thanks!

- Cheng-Ju
  - L2TS and Sparky/Run Control
- Burkard
  - HOLA and PULSAR procurement/setup
- Frans and Sakari
  - SVT-Tx/Rx firmware
- Tomi
  - SVT-Tx latency