

# The NSCL Readout System

- Ron Fox
- Eric Kasten
- Chase Bolen
- Jason Venema
- Max Walters

# Project Goals

---

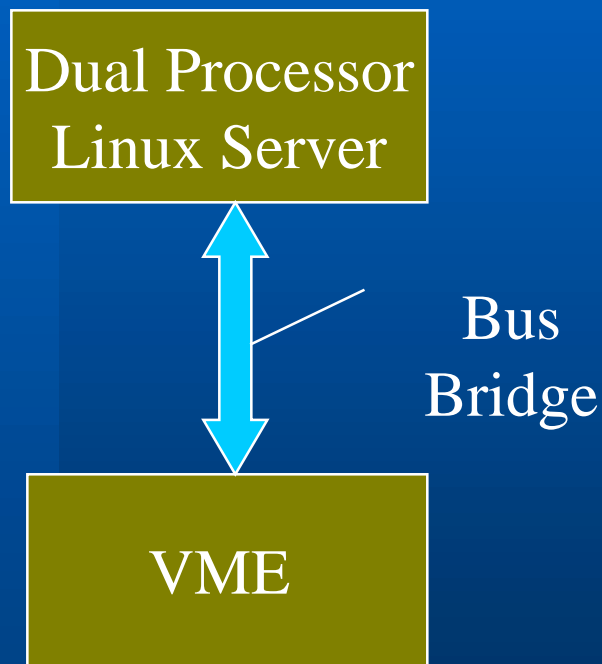
- **Use Commercial Off the Shelf Hardware**
- **Readout Data Using 'Normal' operating system**
- **Do all Readout With User Level Code**
- **Low Event Response Latencies**

# Technology

---

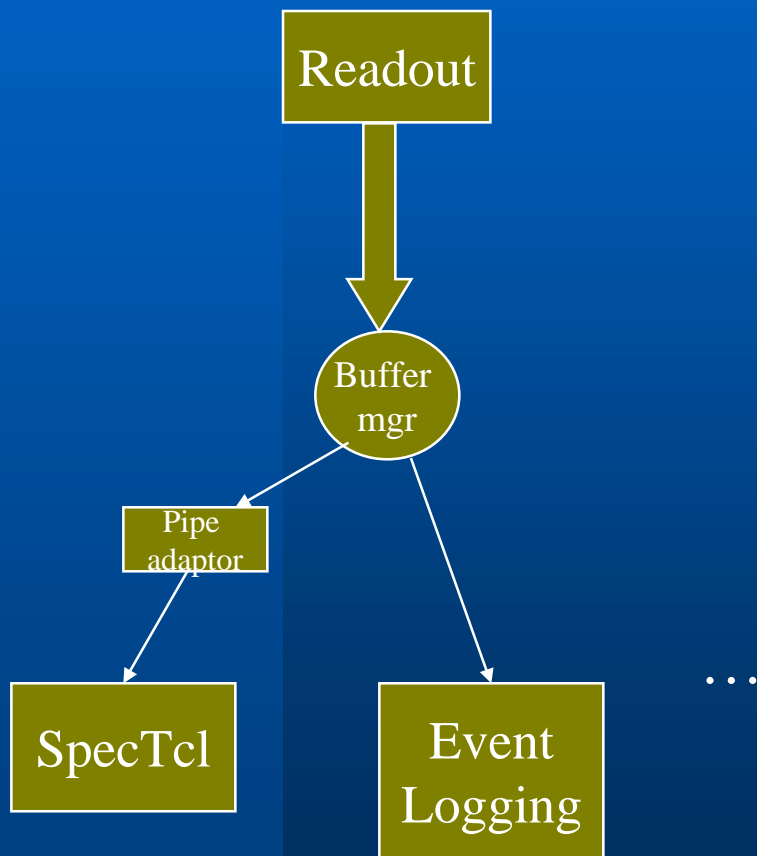
- Commodity Dual Processor PC.
- Bus Bridge PCI ↔ VME.
- Linux Operating System.
  - C++

# Hardware



- **Linux Server e.g. Dell PowerEdge 1550**
- **Bit 3 6xx PCI to VME bus bridge**
  - Supports mmap(2) access to VME address segments.
  - Dma transfers at 35Mb/sec.

# Software

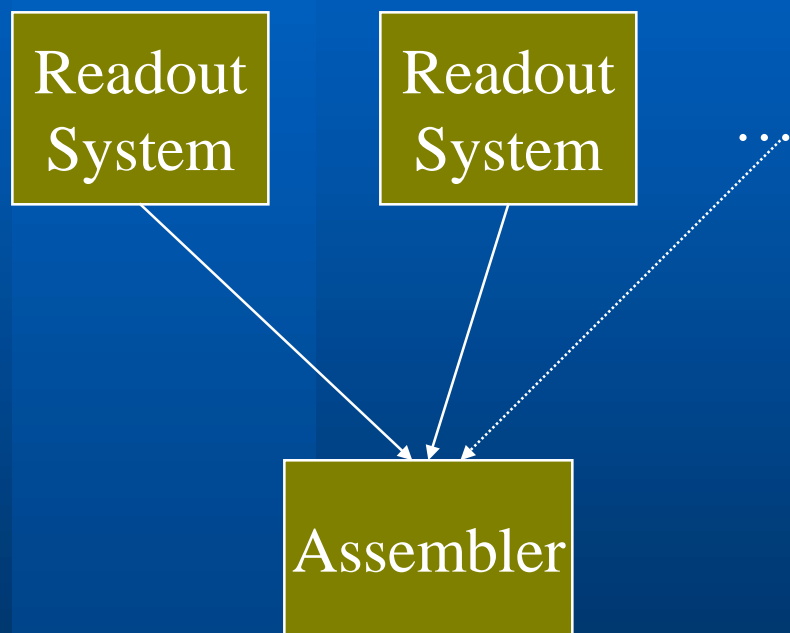


- All Software User Mode C++
- Embedded Tcl/Tk Interpreters.
- Readout mmap s to get to VME, can block read to turn on DMA transfers.

# Readout: Beating Latency

- **Interrupt latencies are  $50\mu\text{s}$  (depending on CPU).**
- **Poll for events:**
  - **Burns a CPU but:**
    - This system is a 'microwave' not a general purpose system.
    - You've got 2 cpus.
  - **Latencies go like the VME access time (less than  $10\mu\text{s}$ ).**

# Scaling: Distributed Event Building (in development).



- **Common Trigger mode.**
- **Time-Stamped free trigger mode.**
- **Assembly in Block-Mode.**

# Other hardware:

---

- **SIS 3100/1100 bus bridge.**
- **Wiener PCIVME**
- **Wiener CC16 with either**
  - **VME interface**
  - **CAMAC interface**