

# Event Building and Dispatching

Mario Cromaz  
*GRETINA Software Working Group Meeting*  
*June 22, 2004*

## Introduction

**Why?** Unlike Gammasphere, the conceptual design for GRETINA does not have prompt hardware event building

- GRETINA digitizers run independently each segment
  - 4 10-channel boards per crystal
  - 120 boards in all
- Events are buffered on each digitizer board for efficient DAQ operation
- Rate varies for each segment  
→ lose time correlation
- Events rebuilt using global (10ns, 48-bit) timestamp associated with each segment event

## 2-step Event Building

### Crystal Event Build

- Occurs in the crate controller
- Time correlates segment events from a given crystal
- Forwards all segment events for a given timestamp range to a decomposition node

### Global Event Build

- Occurs after signal decomposition but before tracking
- Runs on one or more dedicated cluster nodes
- Time correlates:
  - Decomposed Ge events (x,y,z, E, t)
  - Auxiliary detector data
- Events for a given timestamp range are forwarded to a tracking node

## Latencies, Timeouts

- Events cannot be built until all events for a given timestamp range reach the event builder
- Incur latencies:
  - variation of rates among segments
  - variation in decomposition times
- Can be accommodated by buffers whose length exceeds maximum latency

### Timeouts

- In general, one stops waiting when an event from a given board arrives with a later timestamp
- What if it never comes (*i.e* the buffer runs out):
  - Rate differential too high (channel in the noise)
  - Dropped packets (UDP)
  - Channel / board drops out
- Must have a timeout that:
  - Generates an alarm
  - Possibly forces a read or reset of the digitizer card

# Event Dispatching

## addresses:

- which decomposition node processes each segment event
- which tracking node processes each global event

## 2 models:

**push:** sender decides

**pull:** receiver decides (based on local load)

---

Do we need some load balancing scheme?

or

Do large buffers make this matter?

## Summary

- in principle event building is straight forward
- still some open questions:
  - resources required (1 node, several nodes)
  - size of buffers
  - handling of error conditions
  - dispatch model