Real-time use of GPUs for trigger in the NA62 experiment

Felice Pantaleo

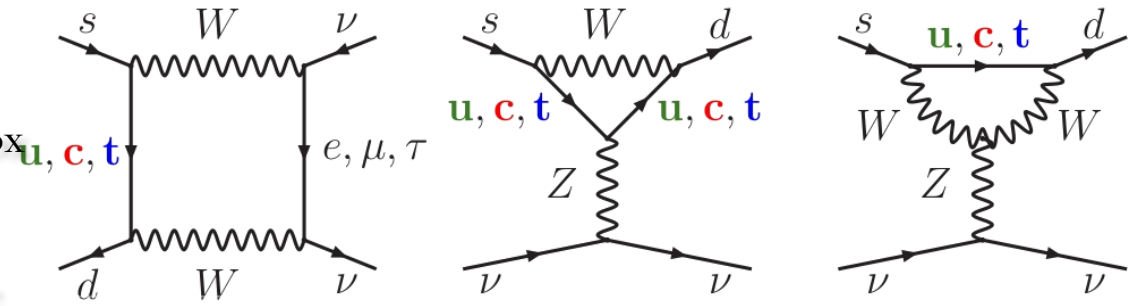
(CERN PH-SFT)

Annual Concurrency Forum Meeting

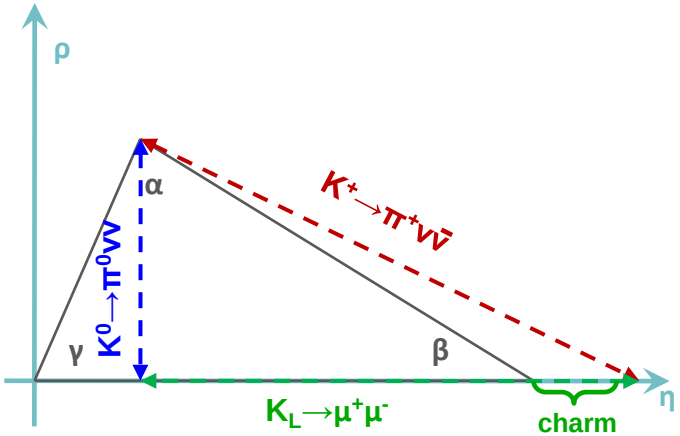
02/05/2013

$K^+ \to \pi^+ \nu\nu$ in the Standard Model

- FCNC process forbidden at tree level
- Short distance contribution dominated by Z penguins and box diagrams
- Negligible contribution from u quark, small contribution from c quark
- Very small BR due to the CKM top coupling $\to \lambda^5$

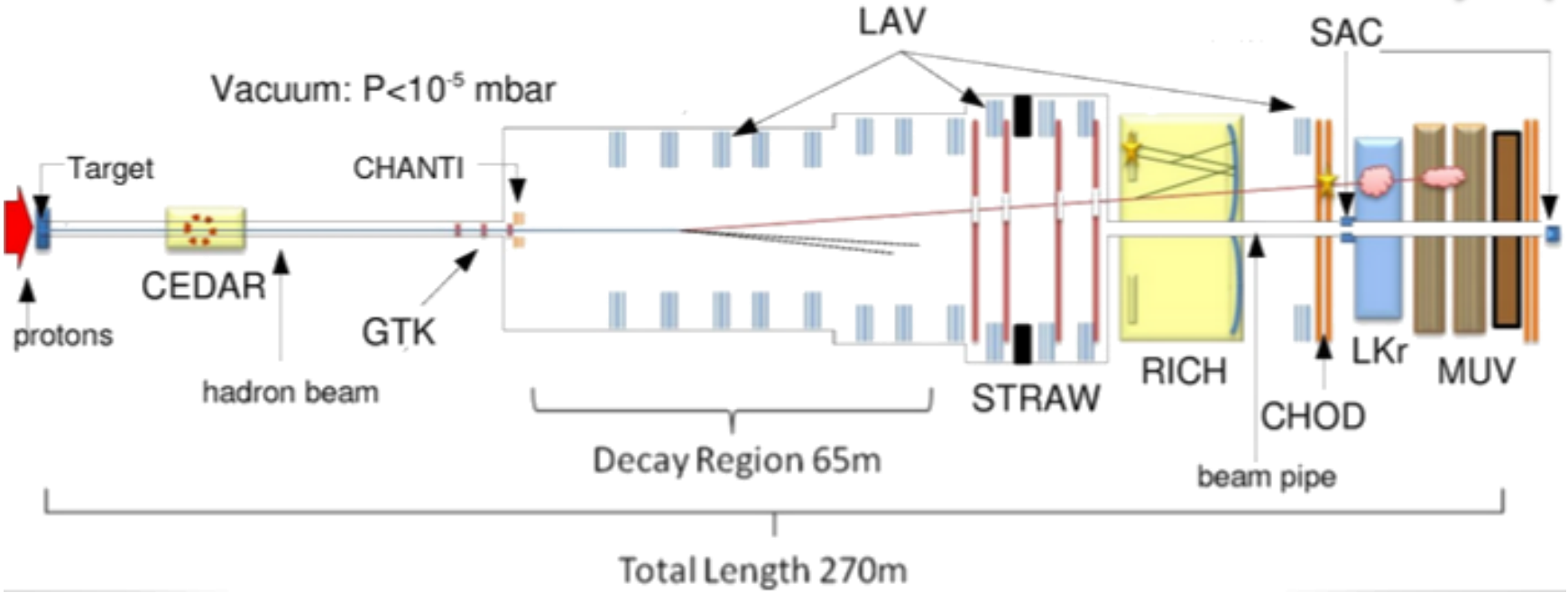


- Amplitude well predicted in SM (measurement of V_{td}) [see E.Stamou]
- Residual error in the BR due to parametric uncertainties (mainly due to charm contributions): ~7%
- Alternative way to measure the Unitarity Triangle with smaller theoretical uncertainty



	G_{SD}/G	Irr. theory err.	BR x 10^{-11}
$K_L \to \pi\nu\nu$	>99%	1%	3
$K^+ \to \pi^+\nu\nu$	88%	3%	8
$K_L \to \pi^0 e^+ e^-$	38%	15%	3.5
$K_L \to \pi^0 \mu^+ \mu^-$	28%	30%	1.5

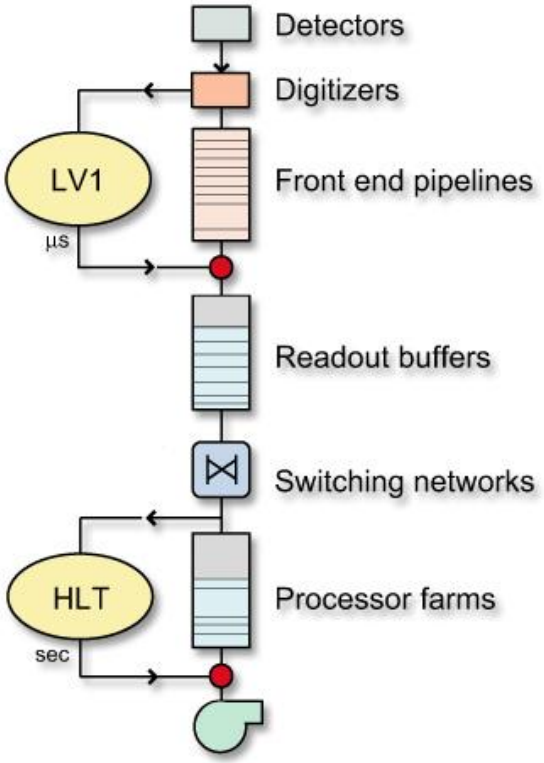
Experimental Technique



- Kaons decay in-flight from an unseparated 75 GeV/c hadron beam, produced with 400 GeV/c protons from SPS on a fixed berilium target
- ~800 MHz hadron beam with ~6% kaons
- The pion decay products in the beam remain in the beam pipe
- **Goal:** measurement of **O(100) $K^+ \to \pi^+\nu\nu$** decays in two years of data taking with % level of systematics
- Present result (E787+E949): 7 events, total error of ~65%.

Generic Trigger Structure

40 MHz
Clock driven
Custom processors

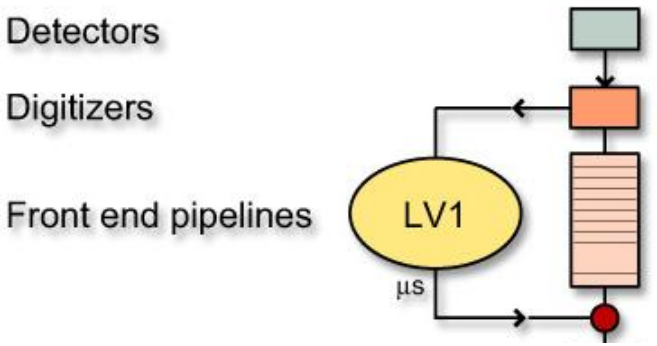


Low Level Trigger
NA62

100 kHz
Event driven
PC network

High-Level Trigger

Low Level Trigger



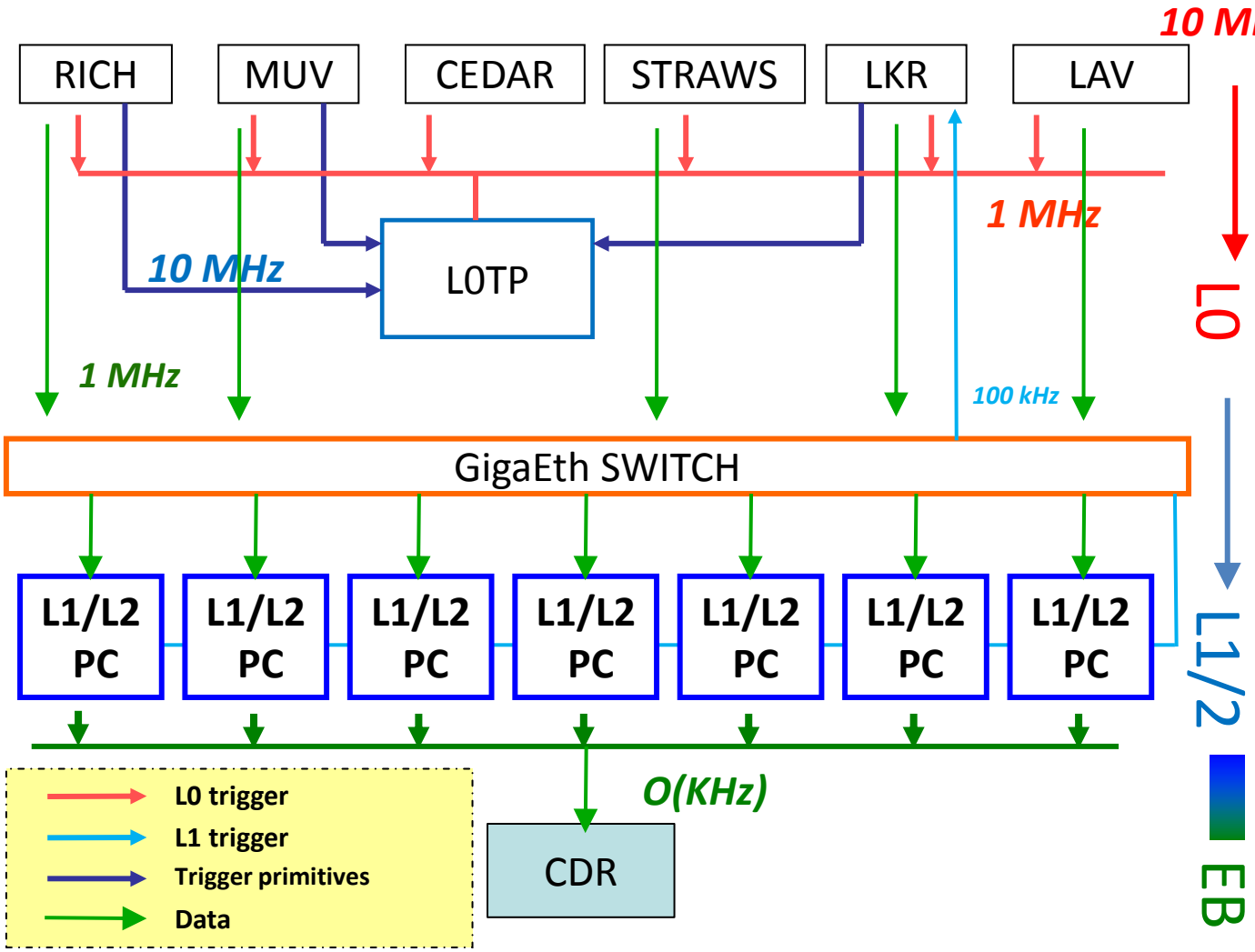
- Time needed for decision $\Delta t_{dec} \approx 1$ ms
- Particle rate ≈ 10MHz
- Need pipelines to hold data
- Need fast response

- Backgrounds are huge
- High rejection factor

- Algorithms run on local, coarse data

- Ultimately, determines the physics

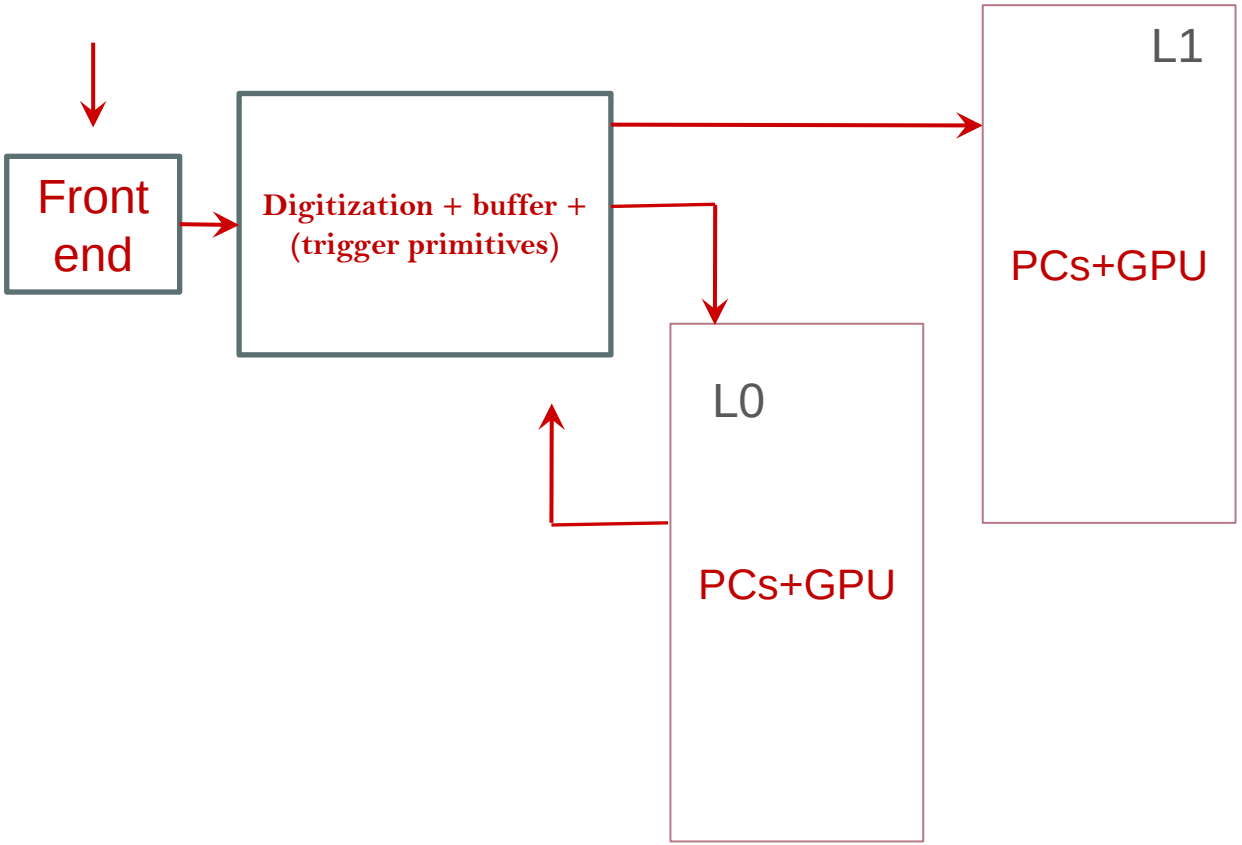
NA62 Trigger



L0: Hardware synchronous level. 10 MHz to 1 MHz. Max latency 1 ms.
L1: Software level. "Single detector". 1 MHz to 100 kHz
L2: Software level. "Complete information level". 100 kHz to few kHz.

GPU as a Level 0 Trigger

- The idea: exploit GPUs to perform high quality analysis at trigger level
- GPU architecture: massive parallel processor SIMD
- "Easy" at L1/2, challenging at L0
- Real benefits: increase the physics potential of the experiment at very low cost!
- Profit from continuative developments in technology for free (Video Games,…)



Data Flow

Max time O(100us)

Read data directly from Network Interface buffers

Filling data structures of arrays

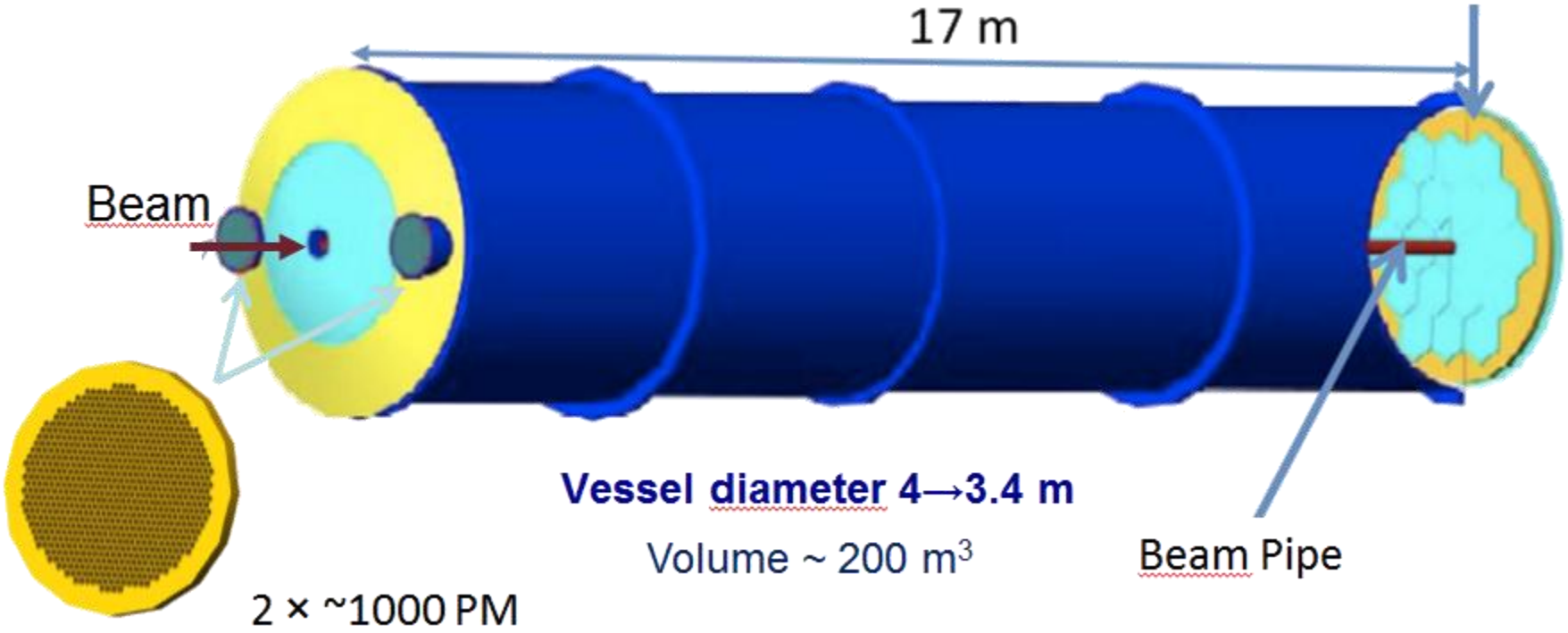
Multiple threads transfer this data to GPU Memory on different streams

Multiple threads launch kernels on different streams

Concurrently transfer the results to the NIC ring buffers and to the frontend electronics

NA62 RICH Level0 Trigger

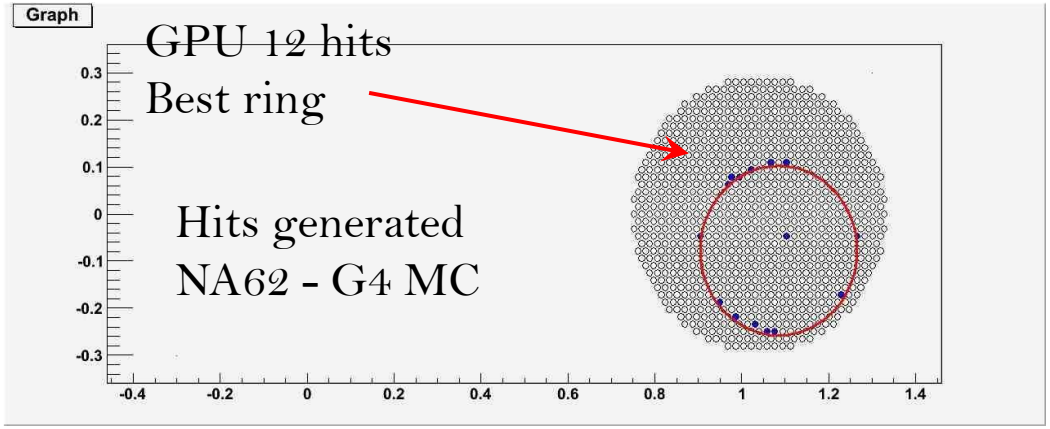
RICH

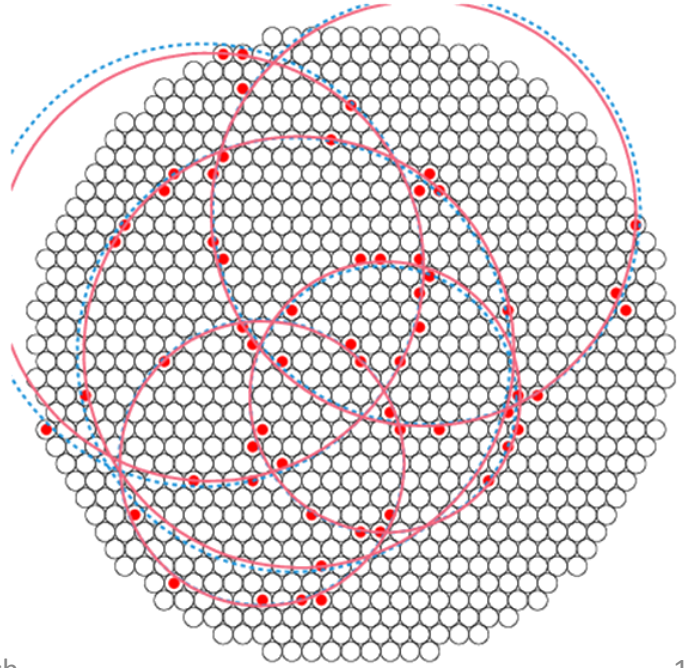


- ~17 m RICH
- 1 atm Neon
- Light focused by two mirrors on two spots equipped with ~1000 PMs each (pixel 18 mm)
- 3s p-m separation in 15-35 GeV/c, ~18 hits per ring in average
- ~100 ps time resolution, ~10 MHz events rate
- Time reference for trigger

Ring Reconstruction

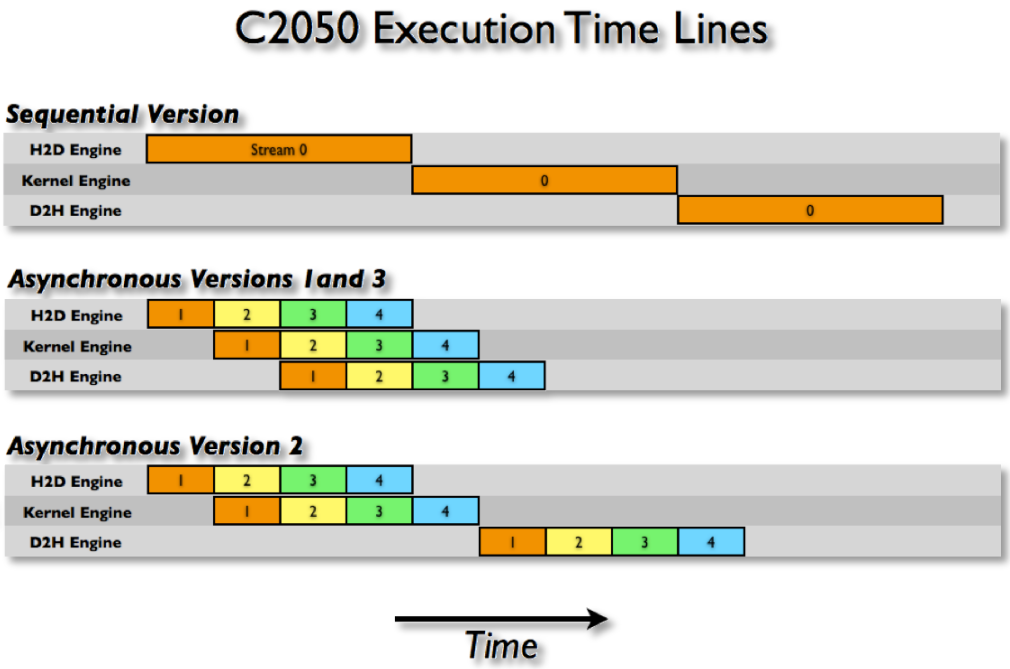
- Natively built for pattern recognition problems
- First attempt: ring reconstruction in RICH detector.





Stream Scheduler

- Exploit the instruction-level parallelism (i.e. **pipelining** streams) to hide latency
- This is usually done by interlacing one stream instructions with another stream ones
- This cannot be done in real-time without the introduction of other **unknown** latencies
- Hybrid CUDA-Pthreads-ntop scheduler implemented to benefit from concurrency at Network – CPU – GPU levels



NA62 RICH Tests

Hardware configuration (1/2)

First Machine

- GPU: NVIDIA Tesla C2050
  - 448 CUDA cores @ 1.15GHz
  - 3GB GDDR5 ECC @ 1.5GHz
  - CUDA CC 2.0 (Fermi Architecture)
  - PCIe 2.0 (effective bandwidth up to ~5GB/s)
  - CUDA Runtime v4.2, driver v295.20 (Feb '12)

- CPU: Intel® Xeon® Processor E5630 (released in Q1'10)
  - 2 CPUs, 8 physical cores (16 HW-threads)

- SLC6, GNU C compiler v4.6.2

Hardware configuration (2/2)

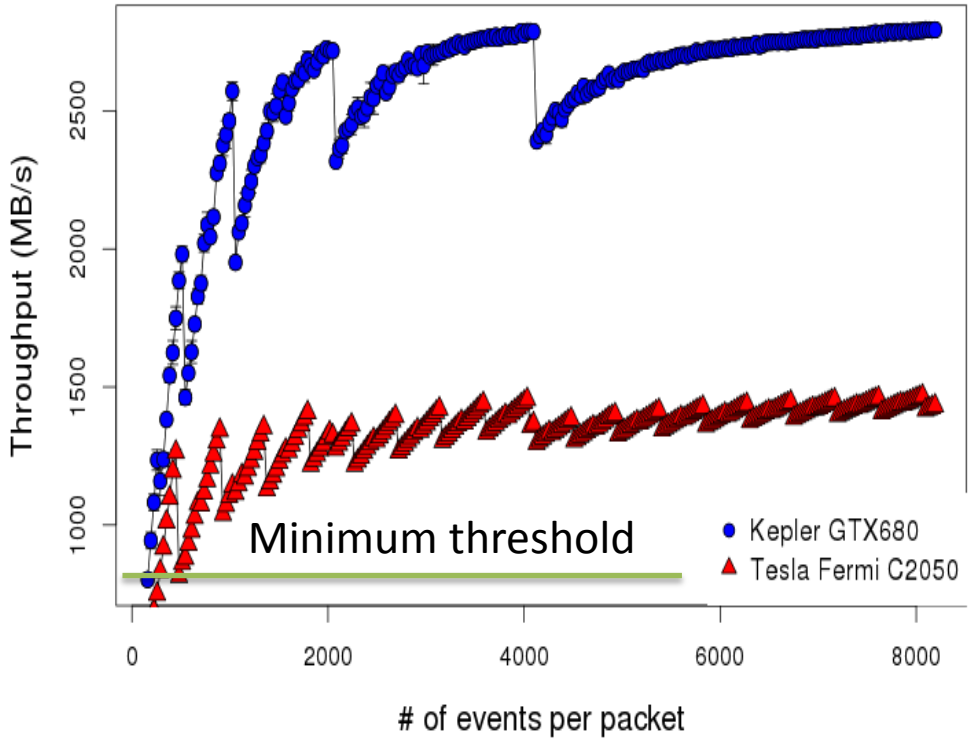
Second Machine

- GPU: NVIDIA GTX680
  - 1536 CUDA cores @ 1.01GHz
  - 2GB GDDR5 ECC @ 1.5GHz
  - CUDA CC 3.0 (Kepler Architecture)
  - PCIe 3.0 (effective bandwidth up to ~11GB/s)
  - CUDA Runtime v4.2, driver v295.20 (Feb '12)
- CPU: Intel® Ivy Bridge Processor i7-3770 (released in Q2 '12)
  - 1 CPUs, 4 physical cores (8 hw-threads) @3.4GHz
- Fedora 17, GNU C compiler v4.6.2

Results - Throughput

The throughput behaviour for a varying number of events inside a packet is a typical many-core device behaviour:

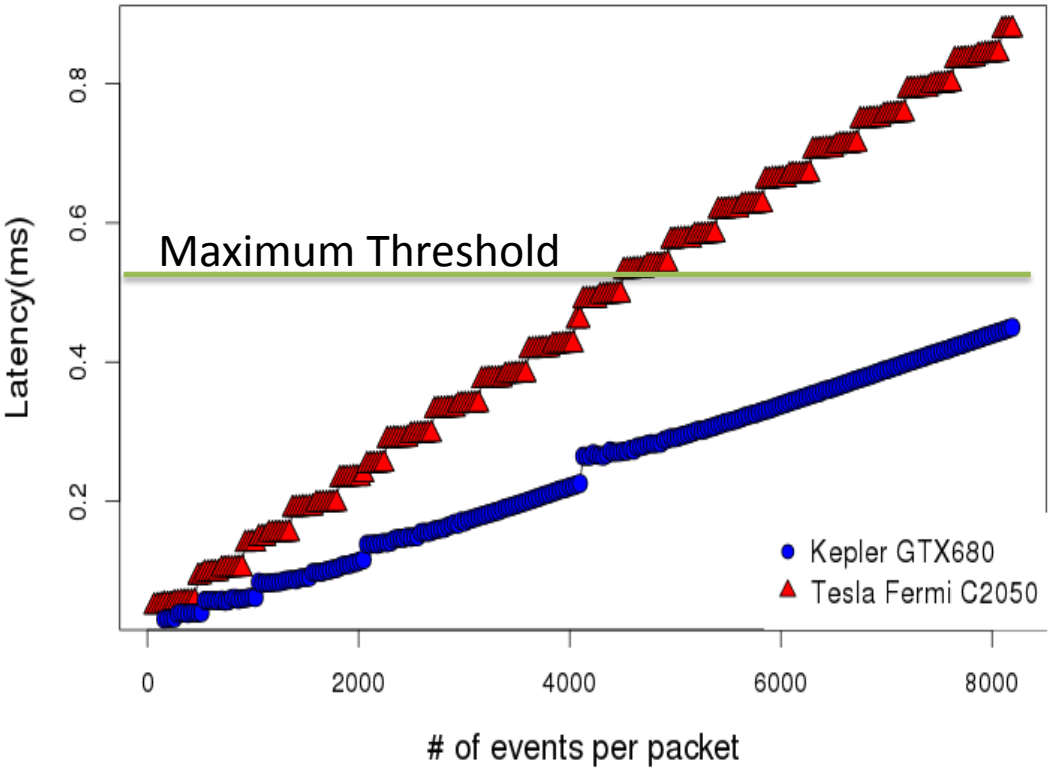
- constant time to process a varying number of events, activating more SMs as the packet size increases
- discrete oscillations due to the discrete nature of the GPU
- saturation plateau (**1.4GB/s** and **2.7GB/s**)



Results - Latency

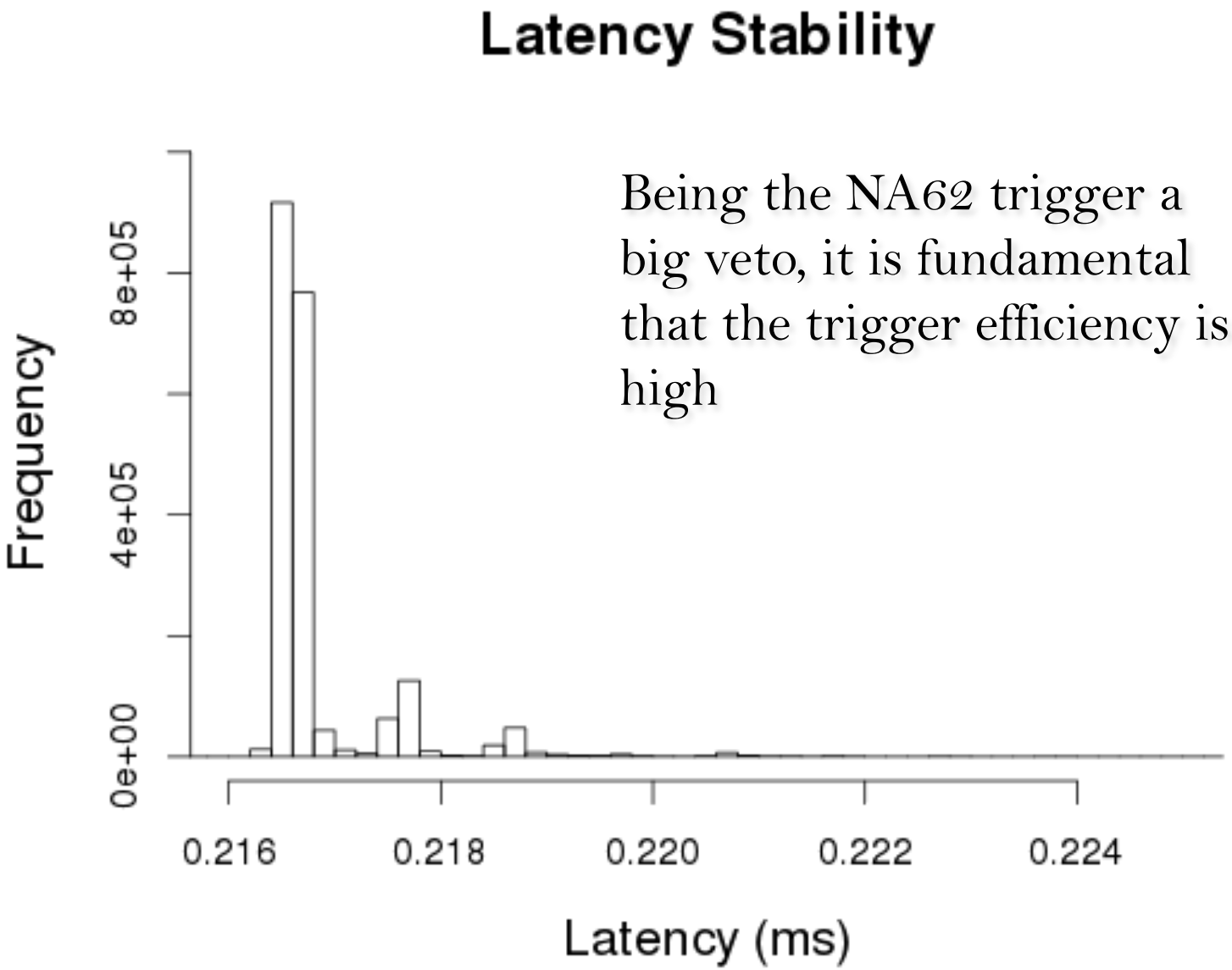
Latency pretty stable wrt event size.

- A lower number of event inside a package is better to achieve a low latency.
- A larger number of event guarantees a better performance and a lower overhead.



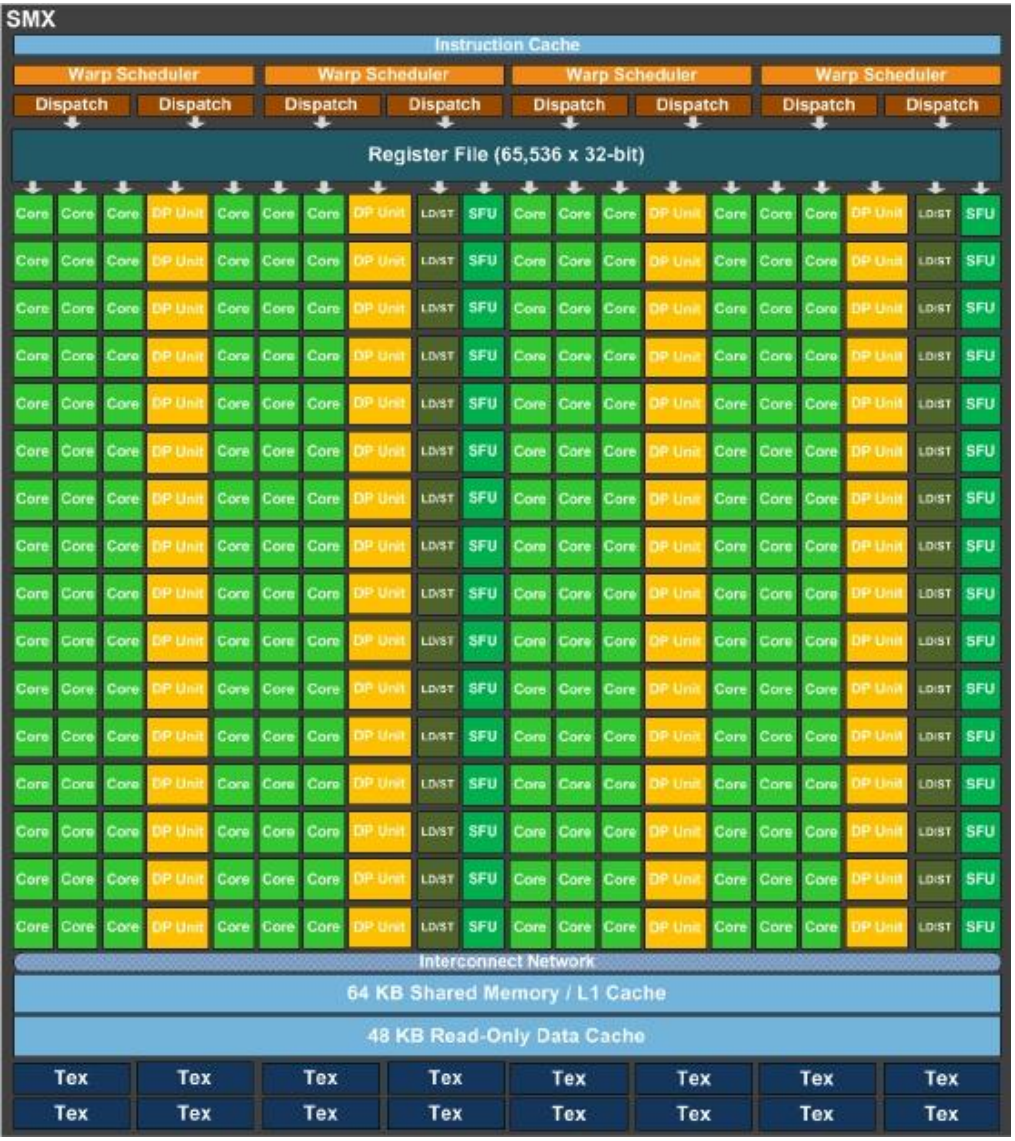
The choice of the packet size depends on the technical requirements.

Results - Latency Stability



Being the NA62 trigger a big veto, it is fundamental that the trigger efficiency is high

CUDA Kepler Architecture



Investigation on which memory to use to store this matrix:

Global memory (read and write)

- Slow, but now with cache
- L1 cache designed for spatial re-usage, not temporal (similar to coalescing)
- It benefits if compiler detects that all threads load same value (*LDU* PTX ASM instruction, load uniform)

Texture memory

- Cache optimized for 2D spatial access pattern

Constant memory

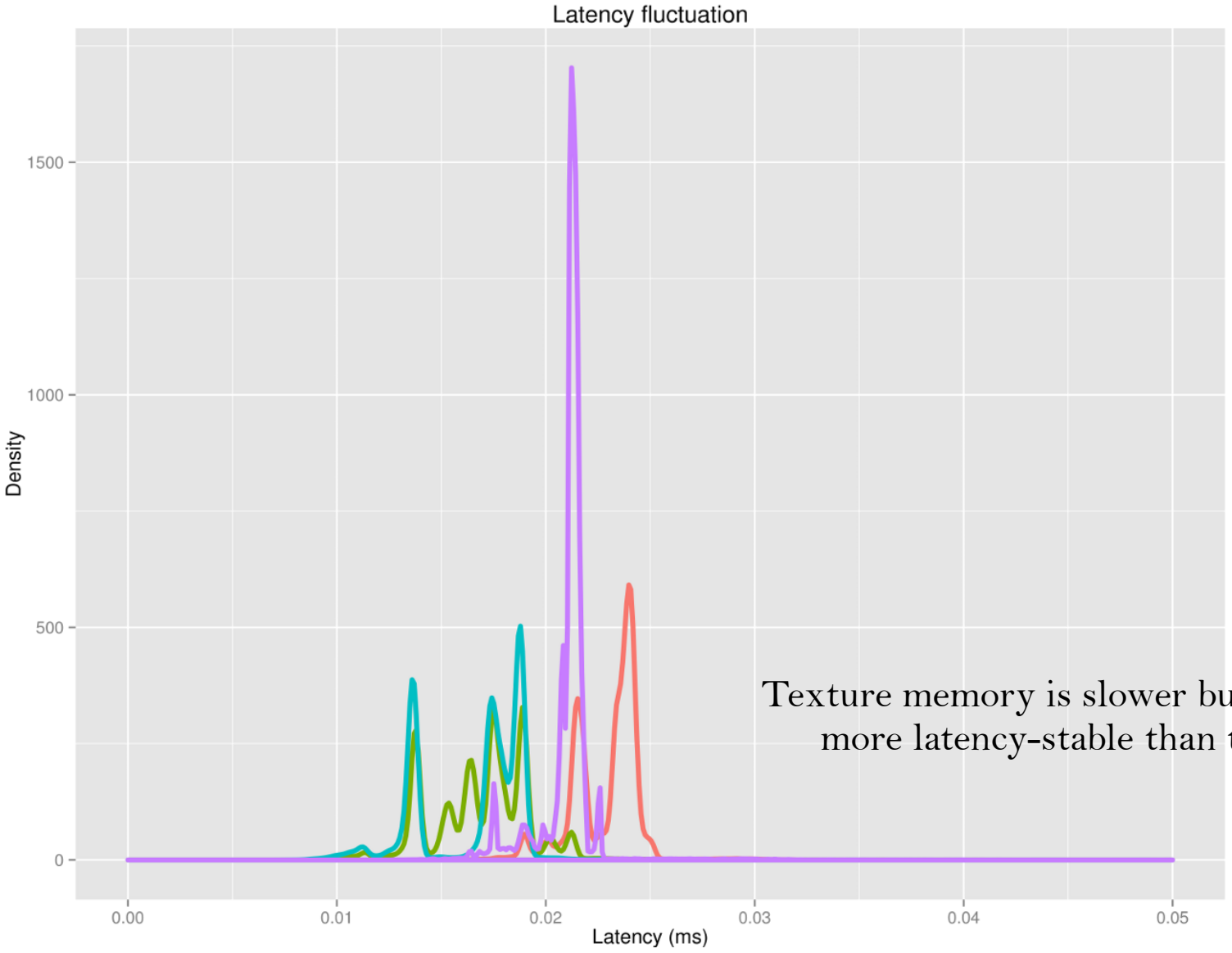
- Slow, but with cache (8 kb)

Shared memory (48kB per SMX)

- Fast, but slightly different rules for bank conflicts now

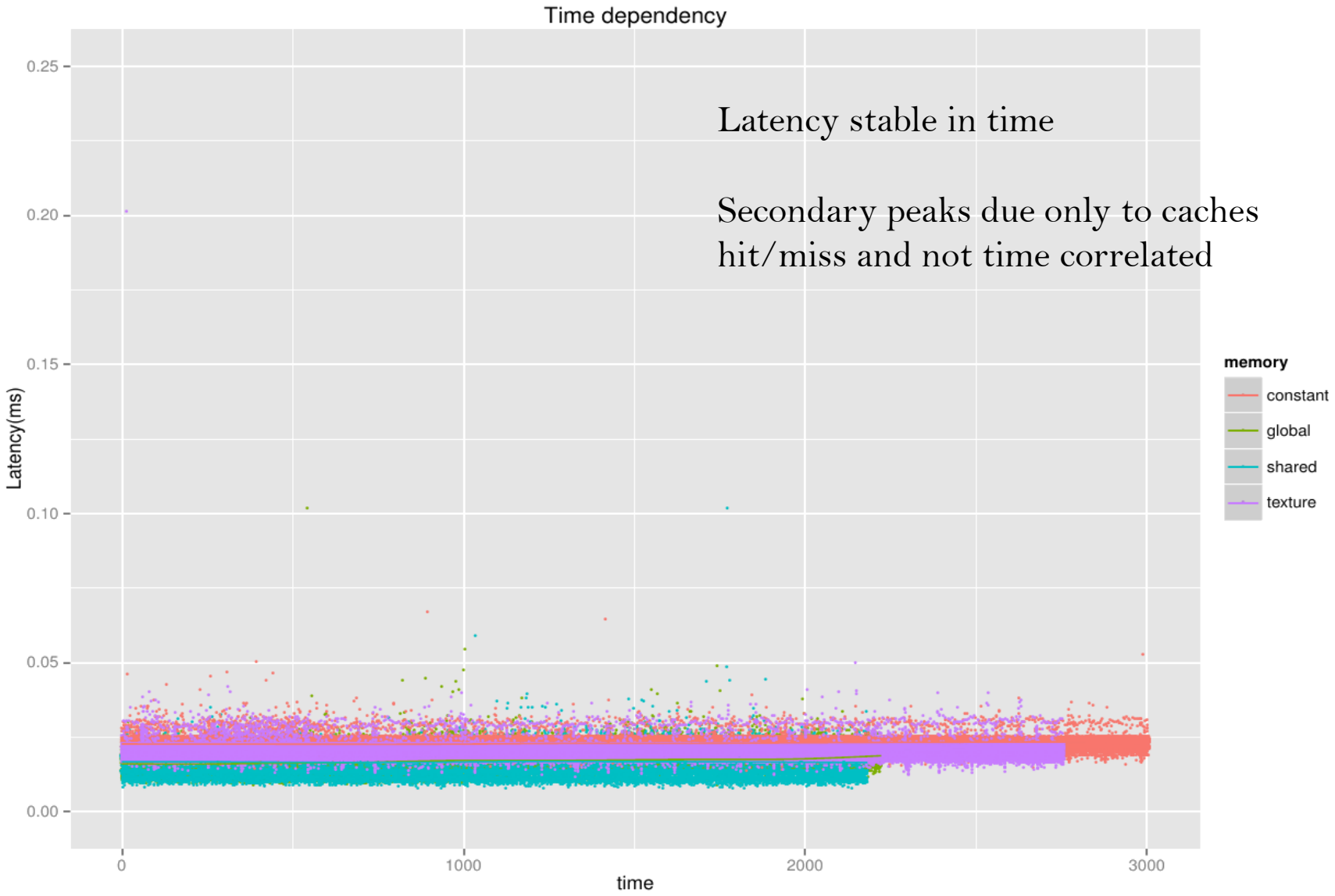
Registers (65536 32-bit registers per SMX)

LUT



Texture memory is slower but way more latency-stable than the others

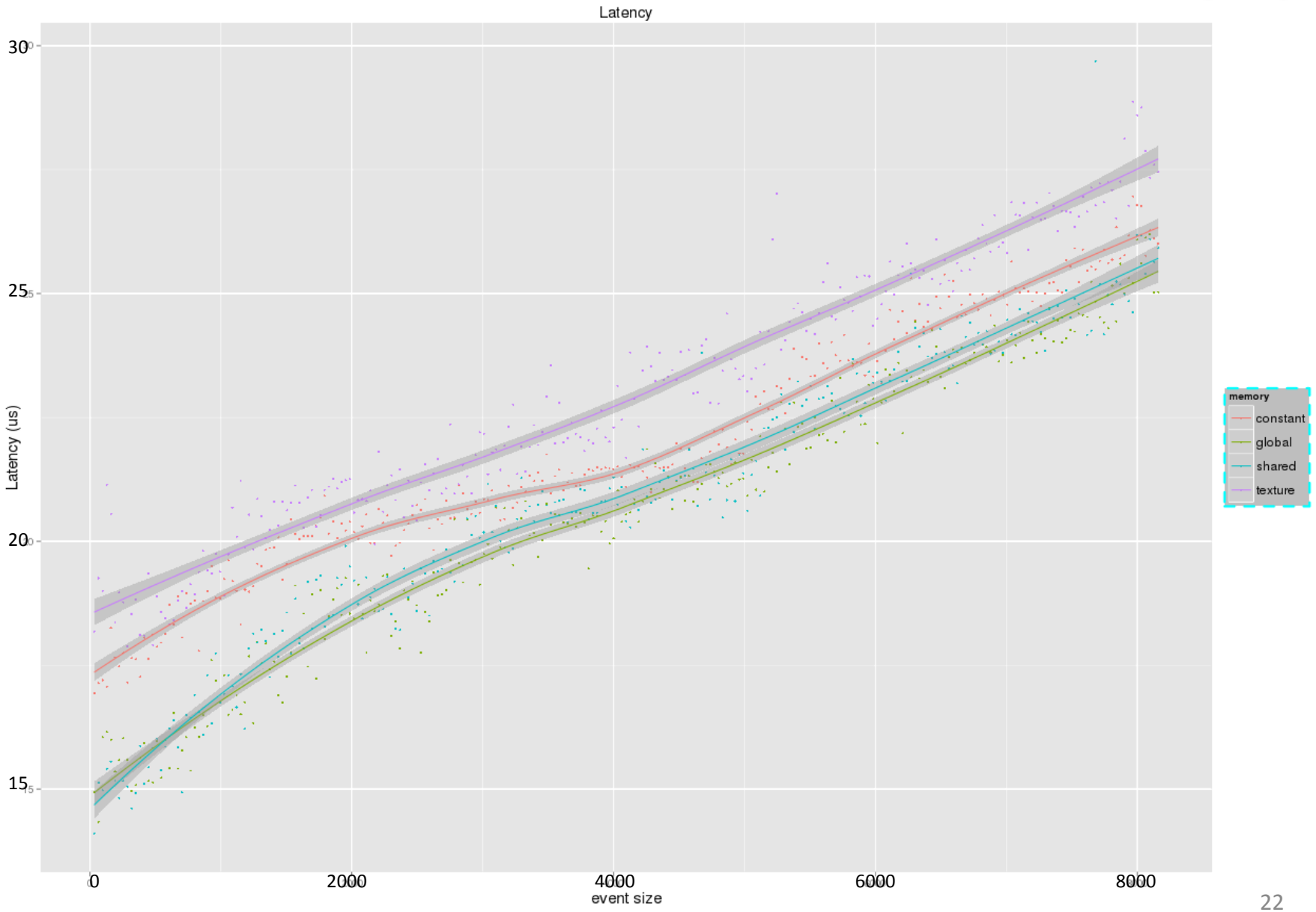
Time dependency



Latency stable in time

Secondary peaks due only to caches hit/miss and not time correlated

Latency


Latency
Latency (us)
30
25
20
15
0
2000
4000
6000
8000
event size
memory
constant
global
shared
texture

Conclusion

- Very specific algorithms written from scratch for the GPU
- A complete system has been tested since the first NA62 technical run in November.
- Setup is not a demonstrator anymore, it is **almost ready for production** phase
- GPUs seem to represent a good opportunity, not only for analysis and simulation applications, but also for more "hardware" jobs.
- Replacing custom electronics with fully programmable processors to provide the maximum possible flexibility is a reality not so far in the future.

Present and Future Work

- Different kinds of synchronization (e.g. external clock, OS alarms, synch between Network Interface and Frontend electronics clocks, etc..) are under evaluation.
- The measure of the trigger response time interval as function will be completed in the next few weeks.