Feeding the Beast: Challenges and Opportunities for Streaming Applications on the Cell B.E Processor

Fabrizio Petrini,
Daniele Scarpazza,
Michael Perrone
Streaming Applications on the Cell/B.E Processor

- **Traditional emphasis on computational performance**
  - Several classes of applications are increasingly limited by the main memory latency/bandwidth
    - Cache size is steadily increasing, but application working sets grow faster
    - We need to carefully optimize the access pattern to the memory hierarchy, in many cases the primary bottleneck
- **With the Cell/B.E. processor we can explicitly “orchestrate” the data transfers between main memory and the local store**
  - Through the Memory Flow Controller (MFC) we can post DMA commands to gather arbitrary segments of memory in local store
  - Extremely powerful mechanism to efficiently support streaming applications
Aggregate Main Memory Bandwidth

![Graph showing aggregate bandwidth vs. block size for numerical and fine-grained applications.](image-url)
Fine Grained Streaming Applications
Hiding Main Memory Latency

Diagram showing the concept of hiding memory latency. The diagram includes:
- DMA transfer stages
- DFA (Data Flow Analysis)
- Transition latency
- Wait periods

The diagram illustrates how longer DMA transfer times are used to hide the latency of memory access.
Fighting the Memory Wall
DMA Performance: Picking the Right Parameters
A Virtual L3 Cache

- By properly streaming memory requests (DMAs) we can implement a virtual L3 cache
  - As large as the TLB coverage of each Processing Element
    - Several Gygabytes
    - With an access latency of 30 nanoseconds
  - Unique opportunity for streaming applications
DMA Transfer Gap vs. Throughput

![DMA Transfer Gap vs. Throughput](image)