

*Presentation to SOS-10 Panel on Data Intensive Computing :*

# Data Intensive Computing through Integrated Memory Structures

Thomas Sterling

Louisiana State University  
and  
California Institute of Technology

March 7, 2006





# What is “Data-Intensive”

- It could be the challenge of managing mass storage, or:
- It could also be that class of problems that are distinguished from (or the opposite of) “compute intensive”
- Compute Intensive:
  - Computational workloads that emphasize register to register ALU operations on relatively few data for high data reuse
  - High temporal locality
  - High cache hit rate
- Then – Data Intensive:
  - Computational workloads that emphasize load/store memory operations on relatively large data for low data reuse
  - Low temporal locality
  - High cache miss rate



# Data Intensive as Memory Oriented

- High percentage of loads and stores
- Low percentage of data reuse
- Time dependency of computation dominated by memory access time
- Relatively large percentage of global (remote) data access and manipulation
- Predictability of memory access patterns may vary dramatically
  - Spatial locality may vary dramatically
- Two classes of application behavior:
  - Memory bandwidth limited
  - Latency sensitive



# CCT Data Intensive Applications

- Sparse time-varying irregular
  - Sparse matrices
  - PIC codes
  - N-body tree codes
- Directed graphs
  - Semantic nets
  - Data mining
  - World knowledge management
  - Machine intelligence



# New Architecture Solutions

- Accelerators for memory operations
- Processor in memory (PIM) structures
  - Almost two decades old
  - Logic directly connected to row buffers on-memory
  - Exposes up to 3 orders of magnitude in memory bandwidth
  - Reduces latency of access by a factor of 10X
- MIND-2
  - Under development at Louisiana State University
  - Multithreaded for near latency hiding
  - Parcel driven for far latency hiding
- Bandwidth is still the problem
  - Wafer scale integration
  - Direct chip to chip optical interconnects