Musings on Main-Memory Systems (with a DBMS bent)

Guy Lohman
Manager, Disruptive Information Management Architectures
IBM Almaden Research Center
Caveat

• I am not a hardware guy.
• So take these musings with a giant grain of salt.
My Exposure: Blink

• Started in 2007
• Aimed at Business Intelligence queries
• Exploits:
  – Large aggregate main memories
  – Large-scale, shared-nothing parallelism
    • Multi-node
    • Multi-core
    • SIMD (vector operation)
  – SQL operations on dictionary-encoded data
  – Cache-conscious algorithms
• Two accelerator products:
  – z/OS (mainframe) appliance (GA’d Nov. 2010)
  – Informix virtual appliance (GA’d March 2011)
Programming Paradigm Changes

• I thought the trend was toward “programmer oblivion” (Exhibit A: Java)!
• L1 cache is crucial – not getting any bigger!
  ➔ Cache line aware
  ➔ Process groups of rows (“strides”)
• More NUMA aware (core affinity)
• More compiler aware – make sure it…
  • Unrolls loops
  • Exploits vector instructions
• Retrofitting legacy software isn’t likely to be multi-core friendly (efficient)
What’s Coming

• **Lots** more cores
• Probably not commensurate memory bandwidth
• More vectorization (RISC $\rightarrow$ CISC)
• Exploitation of GPUs
• Faster PCI Express
• SSDs $\rightarrow$ Phase Change Memory (PCM)
  – Architectural issues:
    • Additional level in the memory hierarchy? OR
    • Alternative to disk for hot data?
  – DBMS issues:
    • How recover from inconsistent states that PCM preserves?
    • How change data structures and access patterns?
Some Consequences

• (Systems) code will be less portable, not more so!
• Energy consumption and heat are major limitations ➔ ARM and Atom processors
• OLTP & BI converging to Operational Data Store
• Can afford to dedicate some cores to…
  – Monitoring
  – Housekeeping
  – Optimizing
• “Disk is the tape of today.”
  – Mike Stonebraker
  ➔ Memory is the disk of today
  ➔ Cache is the memory of today