

---

# Data-Intensive Scientific Computing: Requirements & Solutions

*Jacek Becla*  
*SLAC National Accelerator Laboratory*

Los Alamos Computer Science Symposium  
10/13/2009



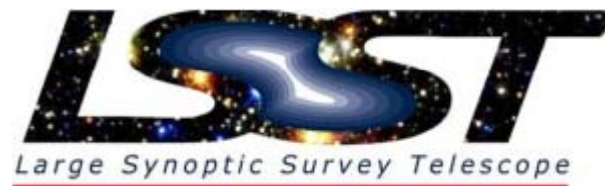
# Outline

---

- Data intensive computing realm
- Complexity of scientific data sets
- Current trends and existing solutions
- Summary

# DISC at SLAC

---



# Analytics Driven

---

- Ingest
  - Controlled, predictable, write-once
- Analyses
  - Ad-hoc, unpredictable, read-many
  - Limiting factor: software and hardware
  - Science – industry: many similarities

# Need to be Flexible, Distributed

---

- Grow incremental
  - Scale out
- Uncertainty, highly varying load
  - System has to adapt, don't want to overbuild
- Large monolithic systems are hard to make failure proof
  - Complexity in H/W vs in S/W

# Sometimes Must be Distributed

---

- Large projects
  - = distributed funding
  - = distributed computing
- Analysis centers of any sizes

# Spindles

---

- 1 PB @50MB/sec = 230 days
- 1 PB in 1h @50MB/sec/disk → 6K disks
- **I/O driven**, not capacity driven
- Can trade some I/O for CPU
  - Compute on the fly
  - Compress (so-so for science data)

# Nodes

---

- Too many disks/node  
= memory bottleneck
- Clusters measured in 100s, 1,000s



# Failures are Routine

---

- Accept it and deal with it
  - Can't disrupt services
  - Must transparently recover
- Avoid shared resources,  
central points of failures

# Other Requirements Imposed by Peta-scale

---

- Pre-execution job cost estimates
- Approx results
  - to speed up exploration
  - to skip failed nodes (if acceptable)
- Job pause/restart
- Self management
  - auto-load balance, auto-fail over, auto-QA
- Relaxed consistency
- Provenance tracking

# Outline

---

- Data intensive computing realm
- **Complexity of scientific data sets**
- Current trends and existing solutions
- Summary

# Order and Adjacency

---

- Time series
- Spatial locality, neighbors

# Multi-D

---

- Typically few dimensions
  - Spatial (2-3)
  - Temporal
  - Sometimes frequency
- Typically one clustering dimension

# Uncertainty

---

- Measurements
- Results

# Outline

---

- Data intensive computing realm
- Complexity of scientific data sets
- **Current trends and existing solutions**
- Summary

# Pushing Computation to Data

---

- Moving data is expensive
- Happens at every level
  - Send query to closest center
  - Process query on the server that holds data



# Improving I/O Efficiency

---

- Limit accessed data
  - Generate commonly accessed data sets.  
Cons: delays and restricts
- De-randomize I/O
  - Copy and re-cluster pieces accessed together
- Trade I/O for CPU
- Combine I/O
  - Shared scans

# Full Data-Set Scans

---

- Sequential access
- No need for indexes
- Simple model

# Architectures in Practice

---

- Off-the-shelf RDBMS based
  - eBay, WalMart, Nokia, BaBar, SDSS, PanSTARRS, LSST
- Custom software, flat files + metadata in RDBMS
  - All HEP, most geo, many in bio, ...
- Custom software, custom format
  - Google, Yahoo!, Facebook, ...

# Distributed Architectures in Practice

---

- Task parallelization models
  - Independent tasks
  - Simple (map/reduce)
  - Complex, full-featured (workflows, shared-nothing MPP DBMS)
- Virtually everybody with PBs is distributed
  - Next stop: cloud

# Convergence

---

- DBMS vendors
  - Rush towards shared-nothing\*
    - Teradata had it, IBM: DB2 Parallel Edition, Oracle: Exadata, Microsoft: Madison
    - Emergence of shared-nothing MPP DBMS startups
  - Adding map/reduce paradigm support
    - AsterData, Greenplum, Teradata, Netezza, Vertica
- Map/Reduce
  - Rush to add db-ish features (schemas, indexes, more operators)

# Spatial Correlations Needed by Many

---

- Science:
  - all geo (solar systems, interplanetary space, solid earth science, atmosphere, ocean, subsurface, water networks, seismic, oil/gas exploration research...)
  - Astronomy
  - bio (e.g., sequences, microscopic and medical imaging)
- Industries
  - oil/gas
  - web companies (mining log data)
  - wall street

But no good solution

# SciDB

---

- Open source DBMS for scientific research
- Shared-nothing MPP DBMS
- Unique features
  - Arrays
    - natively supported arrays (basic, enhanced: ragged, nested...), and array operators
  - Overlapping partitions
  - Basic uncertainty support
  - Executing user defined functions in parallel on independent data

# SciDB – Good for...

---

- Managing / analyzing gridded / n-d data sets
  - Such as images
- Complex analyses on large data sets
  - Time series
  - Spatial correlations
  - Matrix operations
- Designed to scale to 1,000s of nodes



# Summary

---

- Data intensive computing needs **balanced**, shared-nothing, distributed systems
  - It's all about disk I/O, and memory bandwidth
  - Computation centers insufficient
- Big-data users build custom software
  - solution providers rapidly catching up
- Issues with complex spatial correlations not solved
- SciDB – new open source DBMS for scientific analytics

# Related Links

---

- <http://scidb.org>
- <http://www-conf.slac.stanford.edu/xldb07>
- <http://www-conf.slac.stanford.edu/xldb08>
- <http://www-conf.slac.stanford.edu/xldb09>