

# Using Global Behavior Modeling to improve QoS in Cloud Data Storage Services

Jesús Montes, Bogdan Nicolae, Gabriel Antoniu,  
Alberto Sánchez, María S. Pérez

December 2010

Indianapolis (IN), USA



# Introduction

- Cloud Computing
  - Computation as utility
- New possibilities for the large public, such as complex data processing
  - Dryad
  - **MapReduce**

# MapReduce

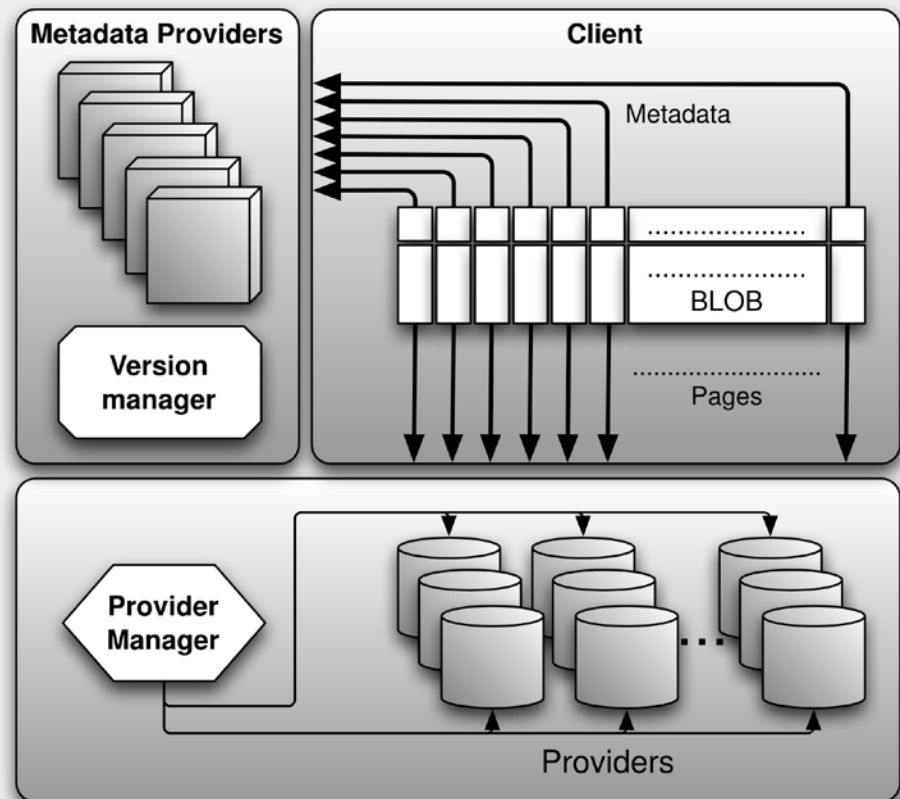
- Platform for large-scale, massively parallel data processing
- Application that can be deployed in the Cloud
- Critical component: Underlying **storage service**
  - Performance: High aggregated throughput under heavy access concurrency
  - **Quality of Service**: Stable throughput for each individual access

# BlobSeer

- Data management for large, unstructured data
  - Very large data (TB) – BLOBs: Binary large objects
  - Highly concurrent, fine-grain access (MB): R/W/A
- Key desing choices:
  - Decentralized data and metadata management
  - Multiversioning exposed to the user
  - Lock-free concurrent writes (enabled by versioning)

# BlobSeer architecture

- Providers
- Provider manager
- Metadata providers
- Version manager
- Clients



# Storage QoS in MapReduce

- Resource failures
- Complex data access patterns
- Complexity of both hardware and software resources

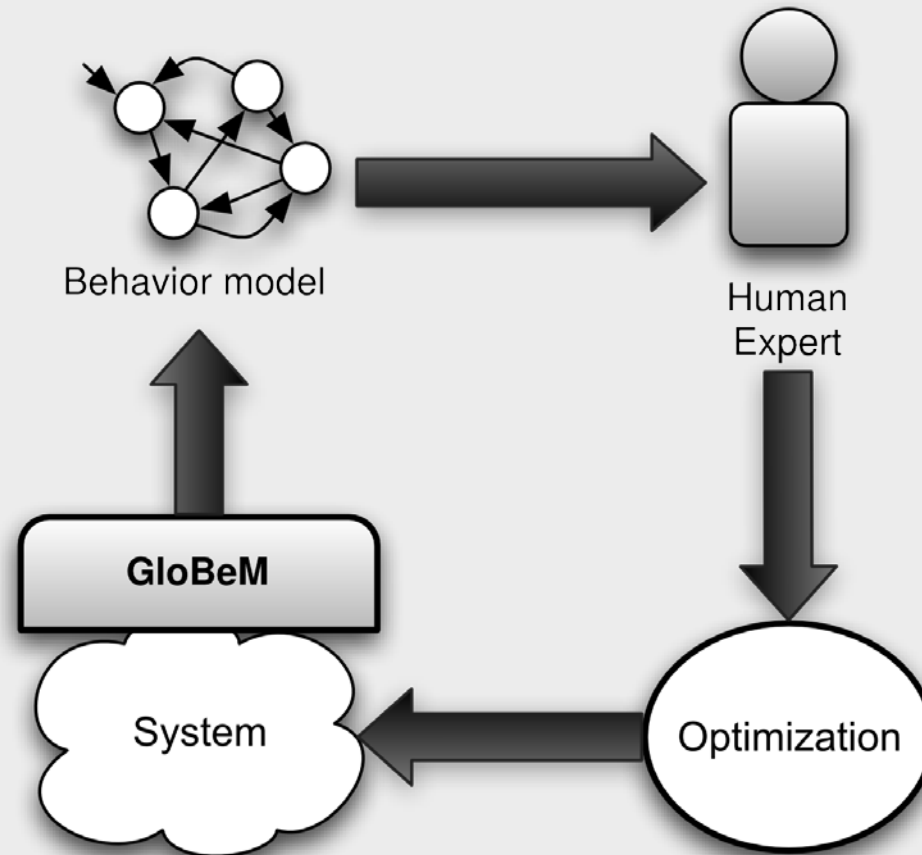
# Proposal

- Automate the process of identifying and characterizing events that have an impact on the storage service QoS
  - In-depth monitoring
  - Application-side feedback
  - Behavior pattern analysis

- Global Behavior Modeling
  - **A Specific methodology to analyze and construct a model of the global behavior of a large-scale distributed system**
- Behavior model characteristics:
  - Finite state machine
  - State characterization based on monitoring parameters
  - Extended statistical information



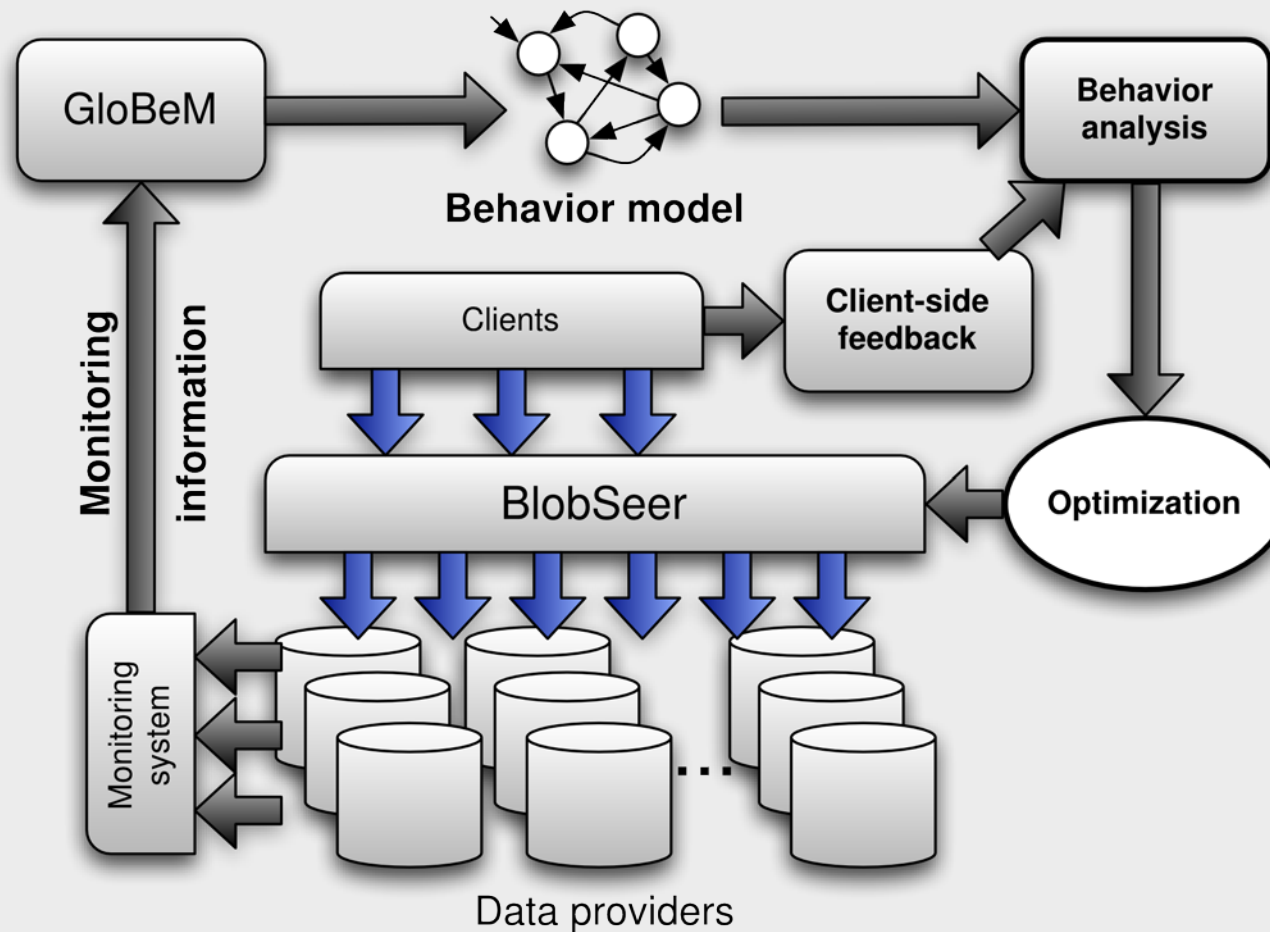
# Behavior modeling cycle



# Proposal

1. Monitor the storage service
2. Identify behavior patterns
3. Classify behavior patterns according to feedback
4. Predict and prevent undesired behavior patterns

# Improving BlobSeer's QoS



# Experimental setup

- Scenario: MapReduce data gathering and analysis
  - Read+Write access pattern (10:1 ratio)
  - I/O+Computation time (1:7 ratio)
- Platform: Grid'5000 resources
  - x86\_64 CPUs, 2GB RAM, 1Gbit/s standard Ethernet
  - Two clusters (130 and 275 nodes)
- Failure injection framework
  - Multi-state resource availability characterization (Rood and Lewis, 2007)

# Experimental settings

- Setting A (Grid'5000 Lille cluster)
  - 130 nodes
  - Total of 11TB accessed (1.3TB written, rest read)
  - **Shared resources** for computation and storage
- Setting B (Grid'5000 Orsay cluster)
  - 275 nodes
  - Total of 17TB accessed (1.5TB written, rest read)
  - **Separated resources** for computation and storage

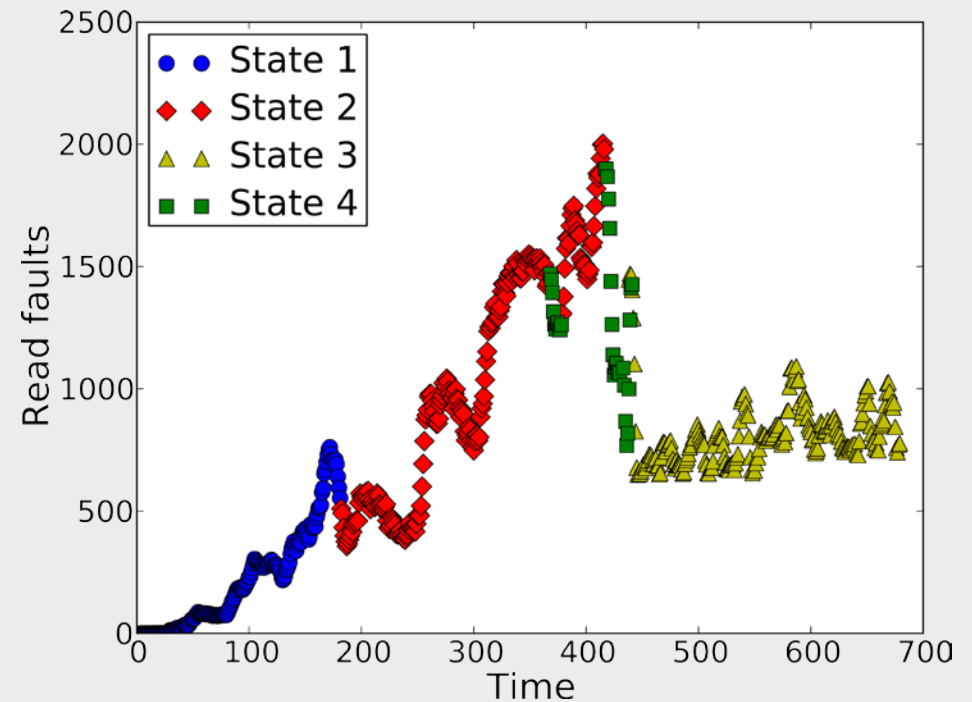
# Experimental procedure

1. Running the original BlobSeer instance
2. Performing global behavior modeling
3. Improving BlobSeer
4. Running the improved BlobSeer instance

# Setting A

- GloBeM identified 4 states

State	Average read BW (MB/s)
State 1	24.2
State 2	20.1
State 3	31.5
State 4	23.9



# Setting A

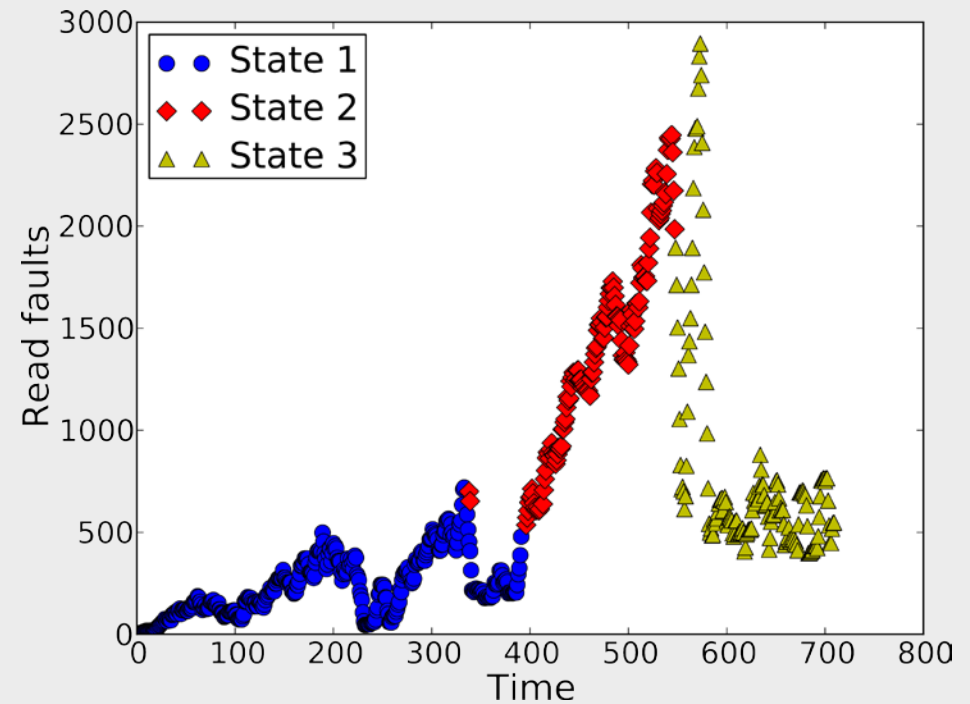
parameter	State 1	State 2	State 3	State 4
Avg. Read ops.	68.9	121.2	60.0	98.7
Read ops stdev.	10.5	15.8	9.9	16.7
Avg. Write ops.	43.2	38.4	45.3	38.5
Write ops stdev.	4.9	4.7	5.2	7.4
Free space stdev.	3.1e7	82.1e7	84.6e7	89.4e7
Nr. of providers	107.0	102.7	96.4	97.2



# Setting B

- GloBeM identified 3 states

State	Average read BW (MB/s)
State 1	50.7
State 2	35.0
State 3	47.0



# Setting B

parameter	State 1	State 2	State 3
Avg. Read ops.	98.6	202.3	125.5
Read ops stdev.	17.7	21.6	21.9
Avg. Write ops.	35.2	27.5	33.1
Write ops stdev.	4.5	3.9	4.5
Free space stdev.	17.2e6	13.0e6	15.5e6
Nr. of providers	129.2	126.2	122.0

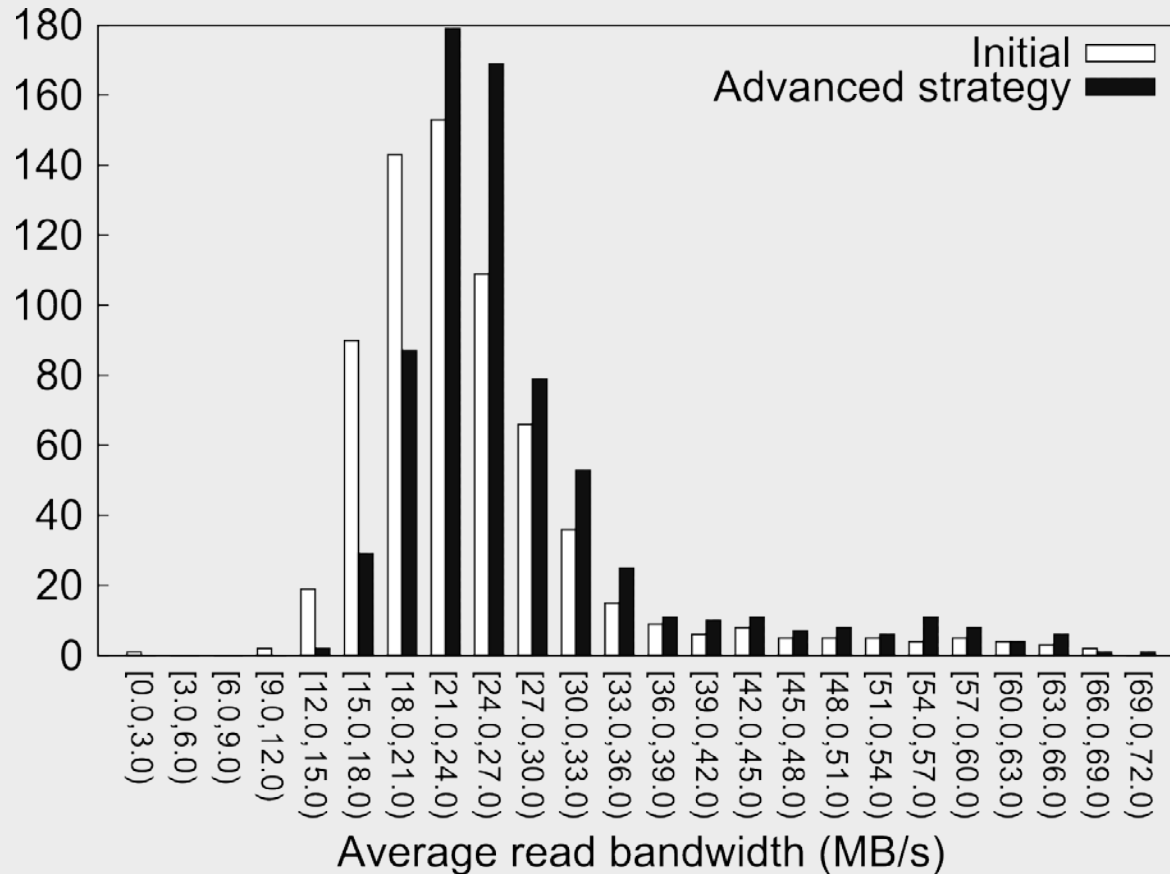
# Analysis

- When a node is dispatching many read operations, a node failure causes many read faults
- Read faults force the client to look for another available replica, reducing the final effective read BW

# Optimizing BlobSeer

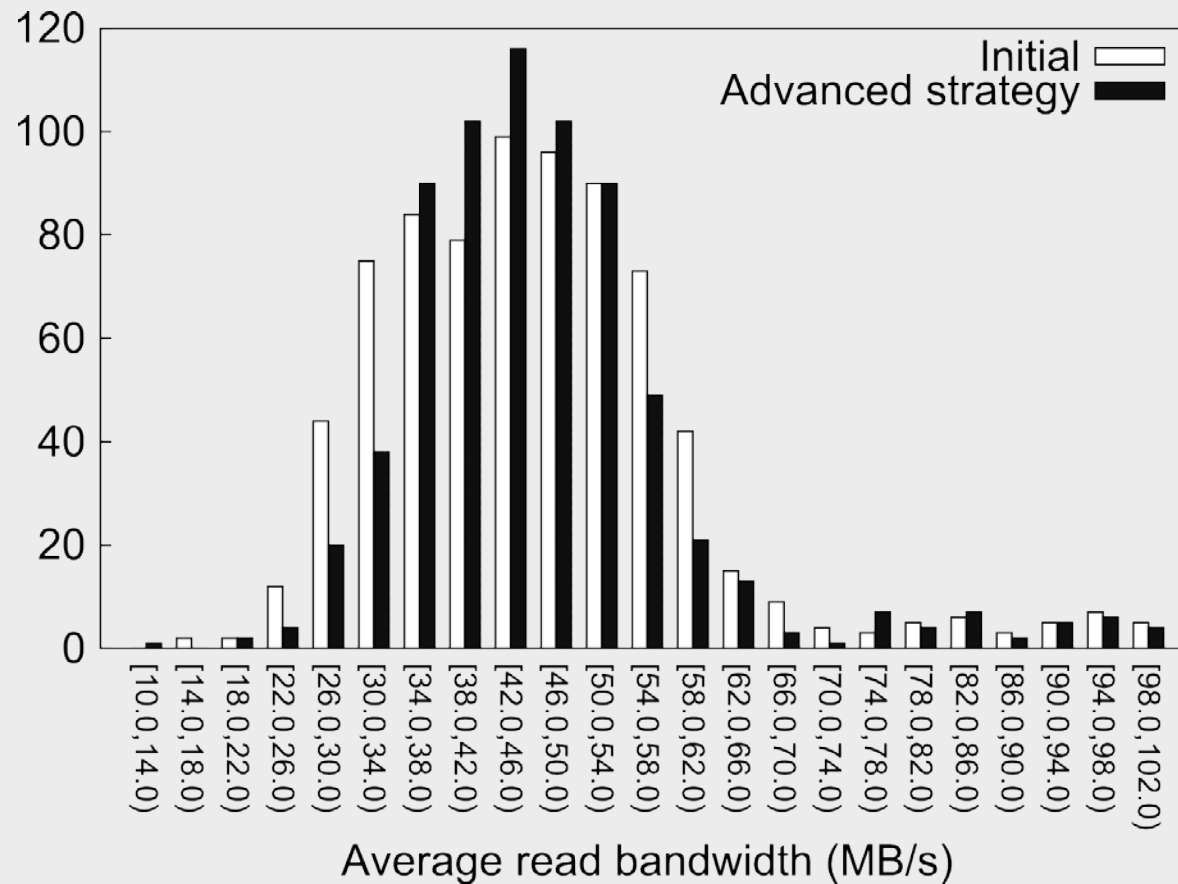
- When, in State 2, selecting a node for writing, reduce priority of those with excessive read operations
  - Reduces the pressure on overwhelmed nodes
  - Improves load-balance and stabilizes read bandwidth, increasing QoS

# Results: Setting A



- Initial configuration read BW avg = 24.9 MB/s, stdev = 9.6
- Advanced strategy read BW avg = 27.5 MB/s, stdev = 7.3

# Results: Setting B



- Initial configuration read BW avg = 44.7 MB/s, stdev = 10.5
- Advanced strategy read BW avg = 44.7 MB/s, stdev = 8.4

# Conclusions

- Cloud storage backend has to ensure stable throughput: QoS constraint
- Our proposal combines component monitoring, application side feedback and global behavior modeling to infer useful knowledge about the storage service
- We have obtained significant improvement in data access QoS

# Thank You!

## Using Global Behavior Modeling to improve QoS in Cloud Data Storage Services

**Jesús Montes**, Bogdan Nicolae, Gabriel Antoniu,  
Alberto Sánchez, María S. Pérez



# Outline

- Introduction
- Proposal
- Experimental Setup and Results
- Conclusions

# BlobSeer

- A back-end for high-level, sophisticated data management systems
  - Highly scalable distributed file systems
  - Storage for cloud services
  - Extremely large distributed databases