

Finding Data in Massive-Scale Storage Systems

Ethan L. Miller

Storage Systems Research Center
University of California, Santa Cruz



Scaling up storage

- ✦ Larger storage systems mean many problems
 - Capacity
 - Performance
- ✦ Critical issue: the ability to *find* data among
 - Billions of files
 - Petabytes → exabytes of data
- ✦ Inability to find data is limiting the utility of large-scale storage systems

Challenge:

scalable search mechanisms

- ◆ Indexing is a critical issue
 - Speed and effectiveness of search limit the usability of very large scale storage systems
- ◆ Very large scale indexes are often resource-intensive
 - Google and Yahoo have web-scale indexes, but they use thousands of processors to do it!
 - Performance is high (memory resident indexes)
- ◆ Indexing can take advantage of locality
 - Users typically aren't searching over the whole file system
 - Users may not have permissions to see everything
- ◆ Challenges
 - Building indexes that scale
 - Building less resource-intensive indexes
 - Building indexes that leverage locality
 - Incorporating security into indexes

Challenge:

gathering metadata for indexes

- ✦ Indexes are only as good as the information that goes into them
- ✦ Critical types of metadata include
 - Content
 - Domain-specific techniques for gathering it
 - May need domain-specific search mechanisms
 - Provenance
 - *How was the data generated?*
 - *On what data and programs does this file depend?*
- ✦ Challenges
 - How can provenance be tracked efficiently?
 - How can domain-specific metadata be handled?
 - Gathered?
 - Indexed as part of the file system?

Challenge:

data mining in mass storage

- ✦ Large storage systems contain a lot of useful data
 - Can be difficult to fully utilize
- ✦ Traditional data mining techniques may not be effective
 - Infeasible to read out the entire storage system for data mining
- ✦ Two potential approaches
 - Index the data when it's written to storage, and use the indexes for mining
 - Distribute computation to the storage devices, allowing them to run in parallel

Challenge:

view-based file systems

- ◆ Traditional hierarchical namespaces are becoming cumbersome
 - Don't scale well: larger & broader hierarchies
 - Ill-suited for many applications
 - Files “belong” in many places in the file system
 - File names are often used to encode data properties
- ◆ Challenges
 - Implementing more flexible namespaces
 - Incorporate attribute-based naming
 - Instantiate directories on-the-fly
 - Allow browsing by many characteristics, not just a single name
 - Integrating flexible naming into the file system
 - Searching should be integrated into file open and lookup
 - Existing solutions (Spotlight, Google Desktop) require cut & paste between “search engine” and actual use of a file

Challenge:

Indexing for exa-scale storage

- ◆ Mass storage systems have major limitations
 - Impossible to read all of the data in a reasonable time
 - Power is a big concern
- ◆ Indexing for mass storage needs to deal with this
 - Traditional approach: large database of stored files
 - Not very scalable for more users or data
 - Vulnerable to failure: rebuilding the DB is close to impossible
 - New approach: distributed index in which each piece of the index covers a portion of the stored data
 - Can scale well to more files and users
 - Actually getting this to work well can be tricky!
 - More reliable: only rebuild the failed part of the index
- ◆ Challenges
 - How can we build an efficient distributed index?
 - How can queries be made efficient?

SSRC research on these challenges

- ◆ The Storage Systems Research Center at UC Santa Cruz is working on some of these challenges
 - Scalable indexing
 - View-based file systems & flexible naming
 - Indexing for archival storage
 - Details at <http://www.ssrc.ucsc.edu/>
- ◆ Many challenges remain!