Optimizing Search in Un-Sharded Large-Scale Distributed Systems

Suraj Chafle[1], Jonathan J. Wu[2], Ioan Raicu[1], Kyle Chard[3]
[1]Department of Computer Science, Illinois Institute of Technology, Chicago, IL
[2]Department of Computer Science, Washington University in St. Louis, St. Louis, MO
[3]Department of Computer Science,` University of Chicago, Chicago, IL

DataSys Data-Intensive Distributed Systems Laboratory

ILLINOIS INSTITUTE OF TECHNOLOGY

THE UNIVERSITY OF CHICAGO

OVERVIEW

- New challenges relating to efficiently discovering, accessing, managing, and analyzing distributed data
- Search framework does not rely on sharding
- Applicable to a range of distributed storage models
- Compares hierarchical index structure

MOTIVATION

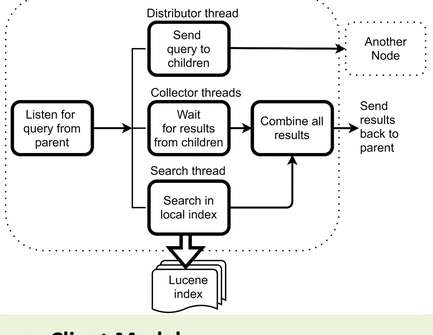
- Storage systems are increasingly distributed
- Discovery and access are crucial for management and analysis of data
- Nodes in unsharded environments more autonomous and network traffic decreased
- No general model for searching in unsharded environments.

OBJECTIVES

- **Search**
  - Discover files based on names and contents
  - Emphasis on speed and scalability
  - Support for near-real-time discovery
- **Environment**
  - Each document remains intact on each node
  - Information stored in system not necessarily balanced among nodes

ARCHITECTURE

- **Lucene**
  - Handles indexing, query processing, searching and scoring of documents
  - Near real time indexing to search capabilities



- **Server-Client Model**
  - Client interface and a server exists on each node
  - Server gets query and begins searching while taking care of query distribution and result collection
- **Query Distribution**
  - Spanning tree constructed using membership list of nodes, allowing for dynamic changes in cluster membership
  - Spanning tree allows queries to be distributed efficiently and reduces network traffic
  - Optimized by sending queries to nodes with larger indexes first, which are more likely to have a longer searchtime

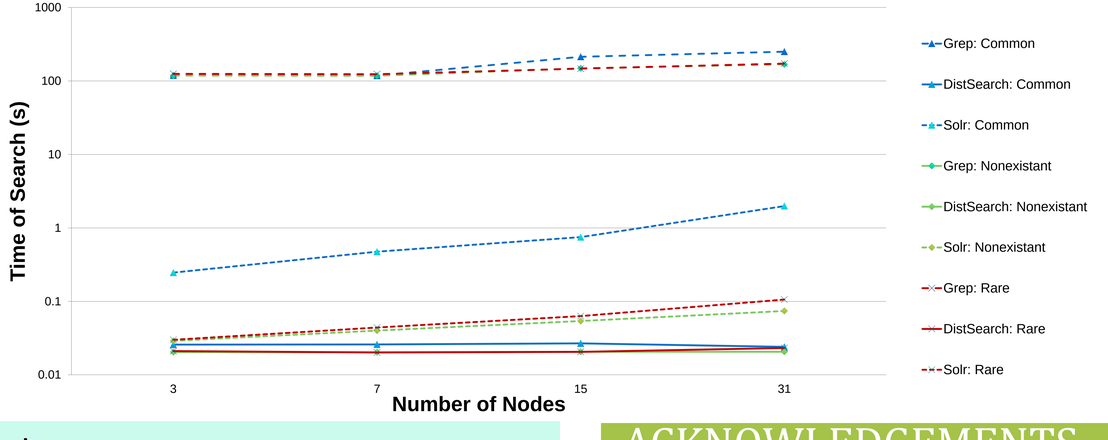
EVALUATION

- **Test Bed**
  - 90000 Wiki documents per m3.large node
  - common, rare, non-existent queries
  - Evaluated against Solr and Grep

FUTURE WORK

- Evaluation with ElasticSearch
- Fault Tolerance
- Smarter distribution structure
- Integration into Globus and FusionFS

RESULTS



- **Results**
  - Lower overhead
  - Faster and scaled better than Solr and Grep

CONTRIBUTION

- Easy to integrate fast, scalable text search for unsharded environments
- Tree-based query distribution model
- Faster search than alternatives when scaled

ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation under awards NSF-1461260 (REU).