Security Aware Partitioning for Efficient File System Search

Aleatha Parker-Wood, Christina Strong, Ethan Miller, and Darrell Long

University of California, Santa Cruz

CRIS
NSF Center for Research in Intelligent Storage

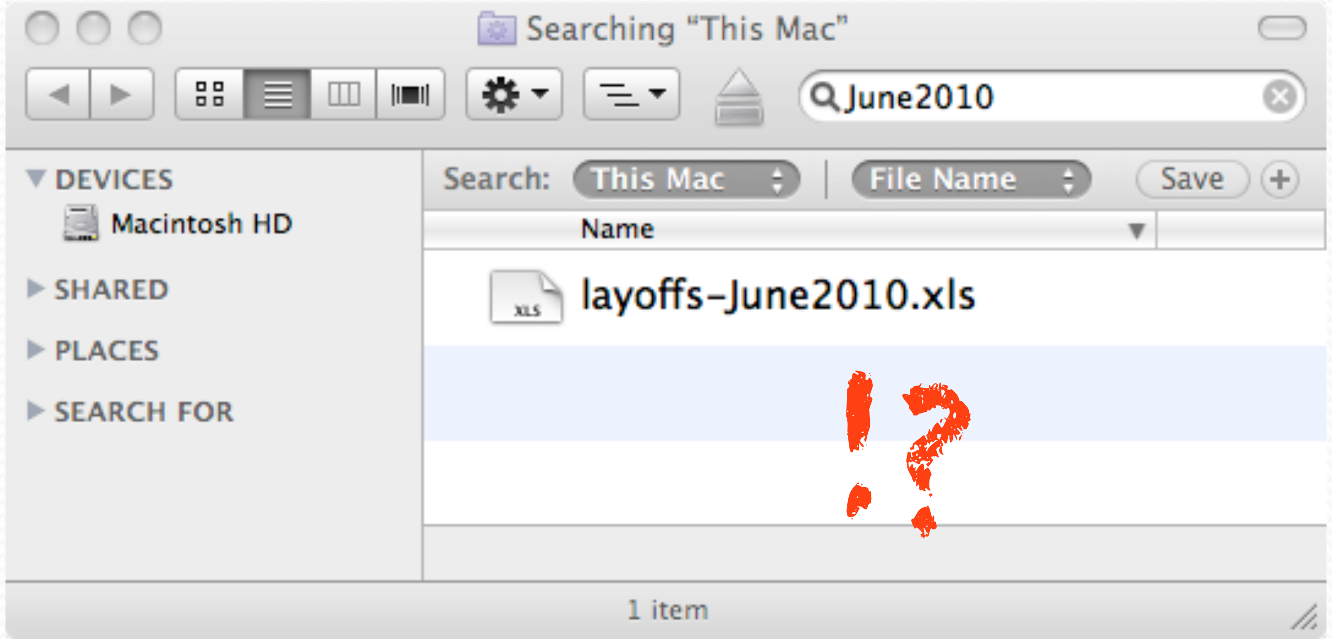
ssrc
Storage Systems Research Center

CITRIS
Center for Information Technology Research in the Interest of Society

Baskin Engineering
UC Santa Cruz

Search and Security

- Security is crucial for file system search
- Users should never see results they cannot see via the file system
- But security is slow
  - Retrieving permissions
  - Filtering



- Ranked search is even harder to secure (Büttcher 05)

Partitioning Can Help

- Partitioning is a popular technique for speeding up search (Leung 09, Hua 09, ...)
- File metadata divided into disjoint partitions
- Each partition has an distinct index
- Bloom filters quickly eliminate irrelevant partitions
  - Probabilistic bitmap
  - One Bloom filter per metadata field
- Search only relevant indexes

Our Algorithm

- We propose a novel partitioning algorithm which directly integrates security
- Fast — O(n)
- Low memory requirements — O(lg n)
- Good properties for efficient search
- Eliminates filtering operations at query time

Contributions/Roadmap

- Security Aware Partitioning Algorithm
- Metrics for Evaluating Partitioning
- Evaluation

Roadmap

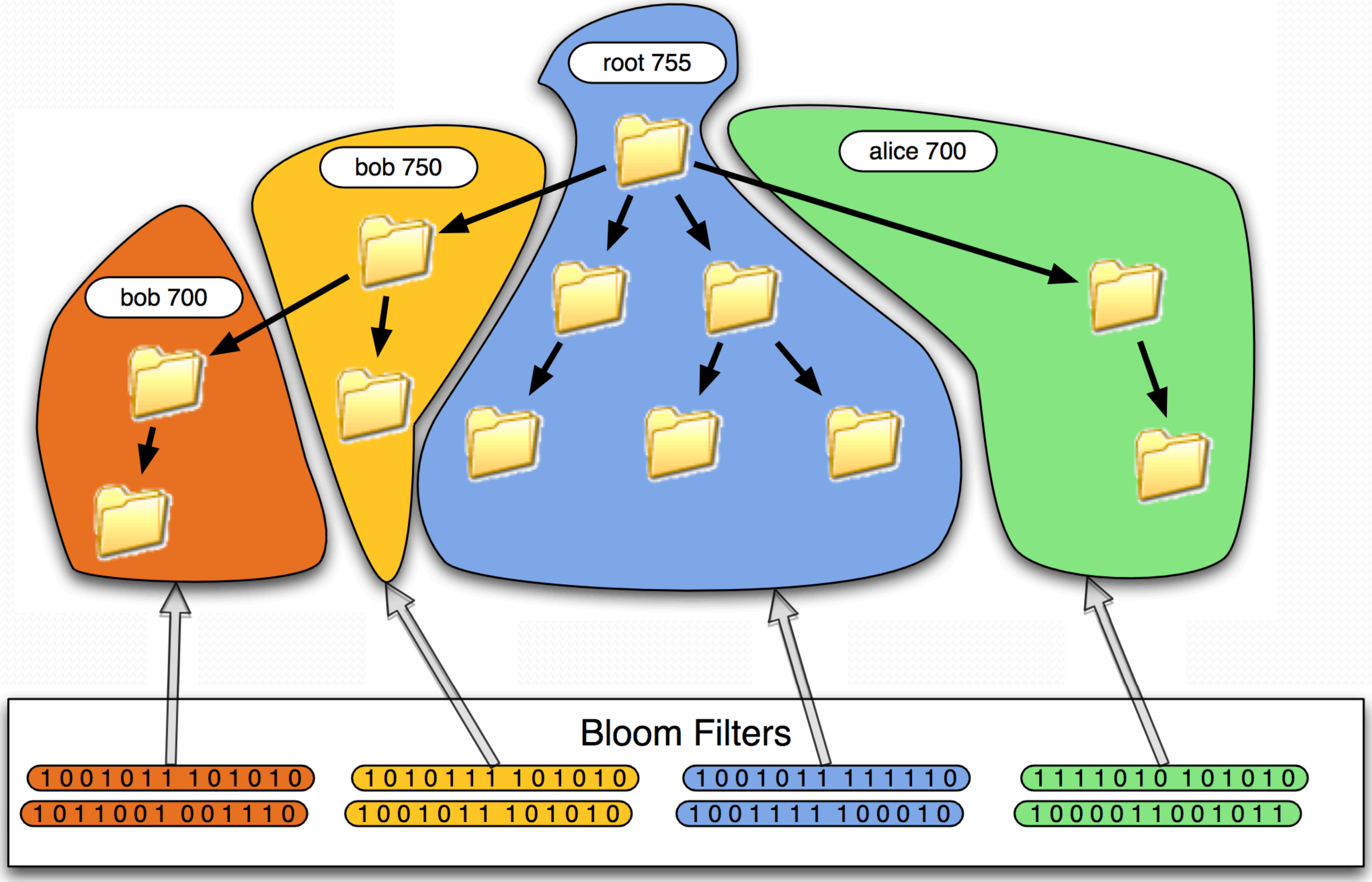
★**Security Aware Partitioning Algorithm**

- Metrics for Evaluating Partitioning
- Evaluation

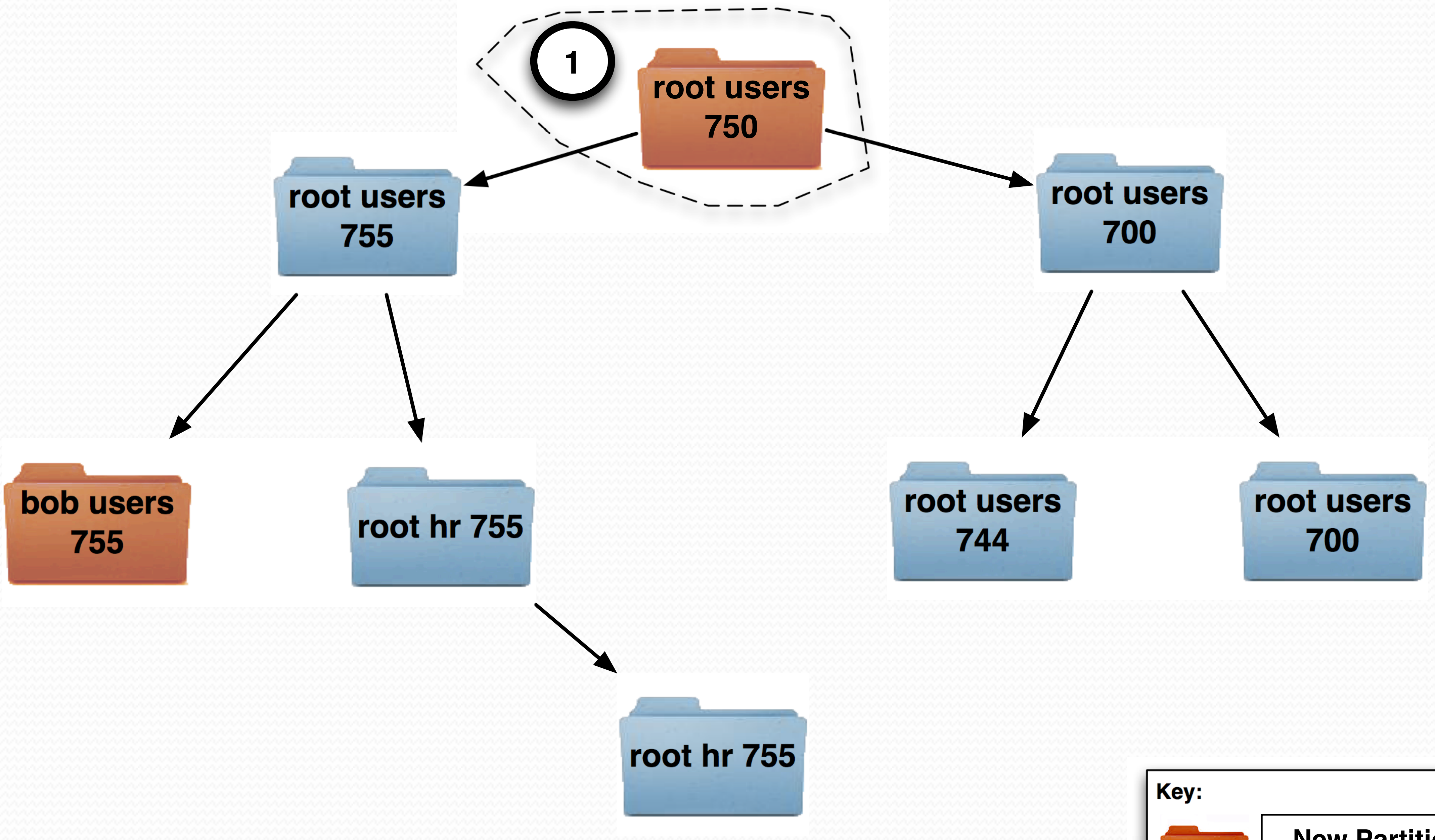
Security Aware Partitioning

- Partition based on security permissions
  - User can see everything in a partition...
  - Or nothing.
- Will use POSIX as example
- Consider read and search for directories, read for files
- Ignore indexes where user doesn't have access
  - No filtering after results are retrieved
  - Retrieves fewer indexes, speeding up search

Security Aware Partitioning



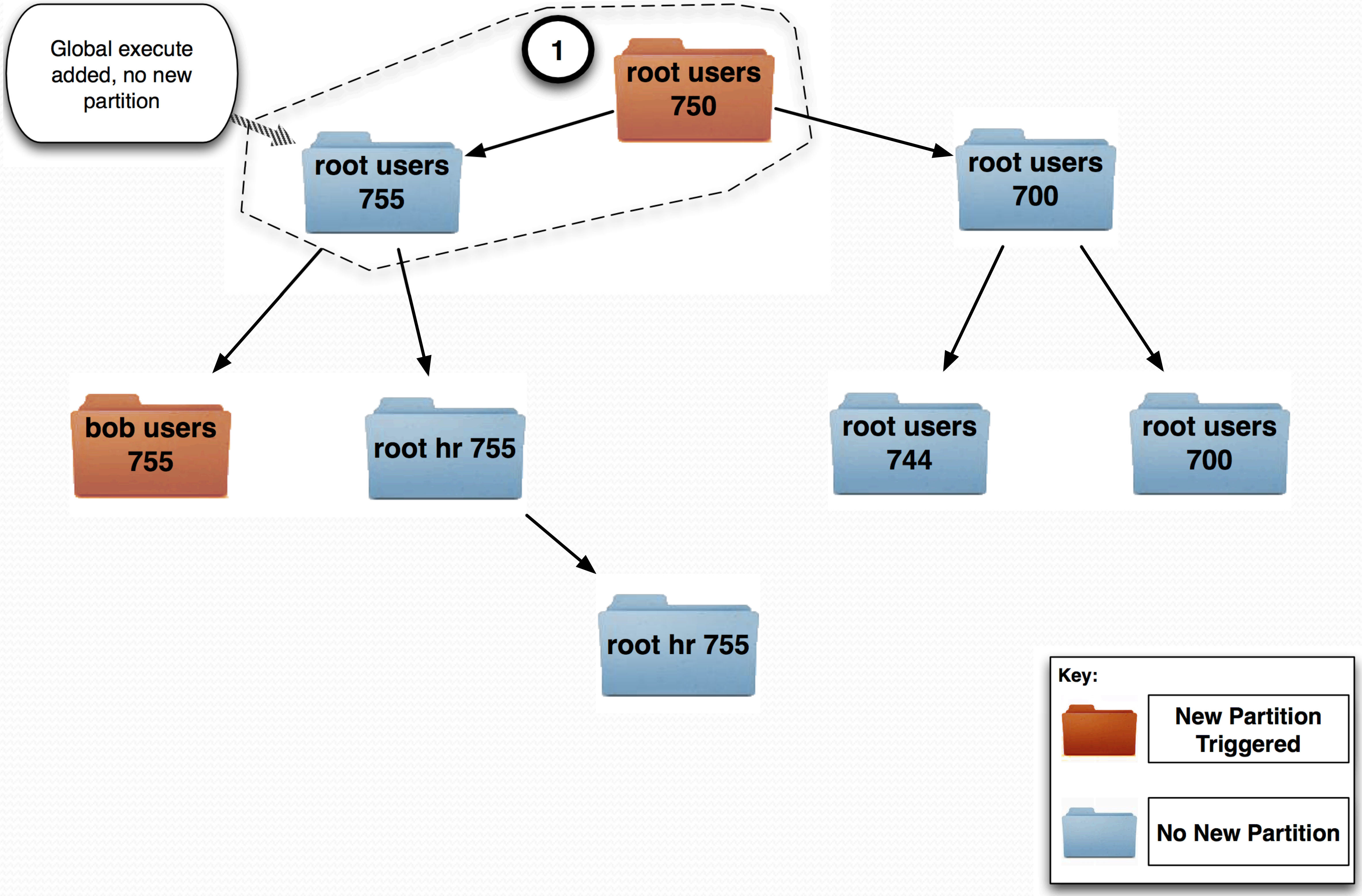
Partitioning Example



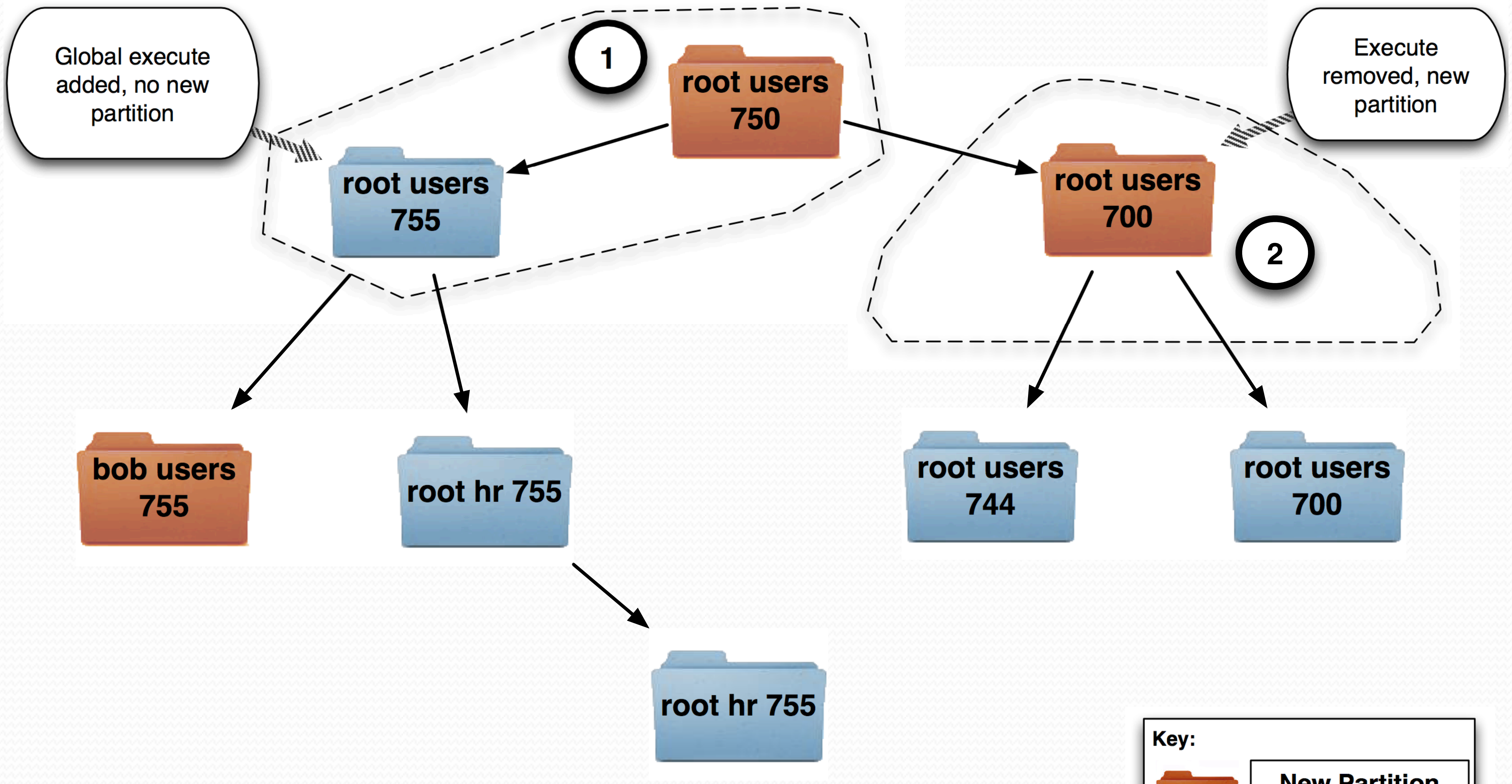
Key:

🟧	New Partition Triggered
🟦	No New Partition

Partitioning Example



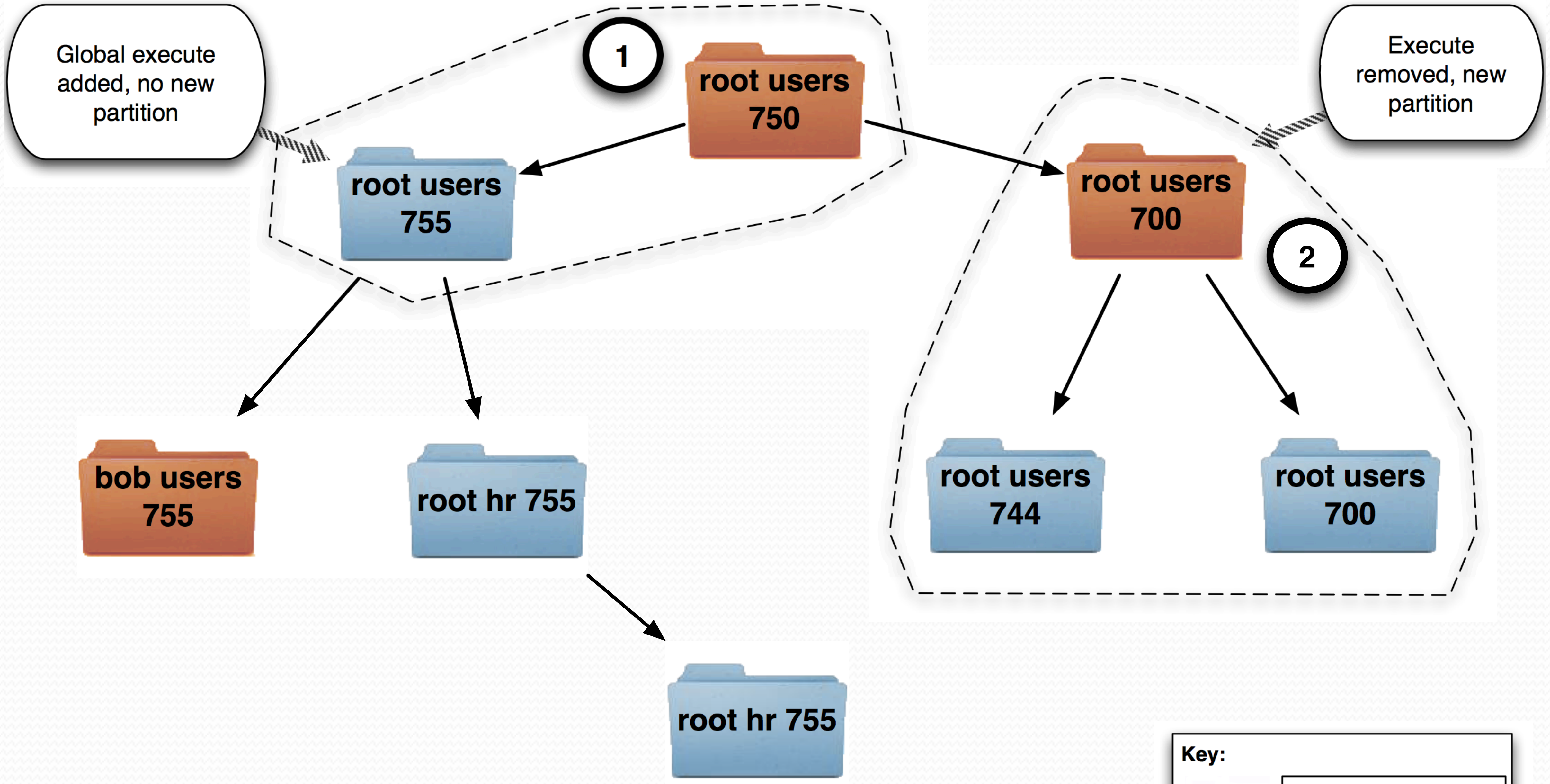
Partitioning Example



Key:

🟧	New Partition Triggered
🟦	No New Partition

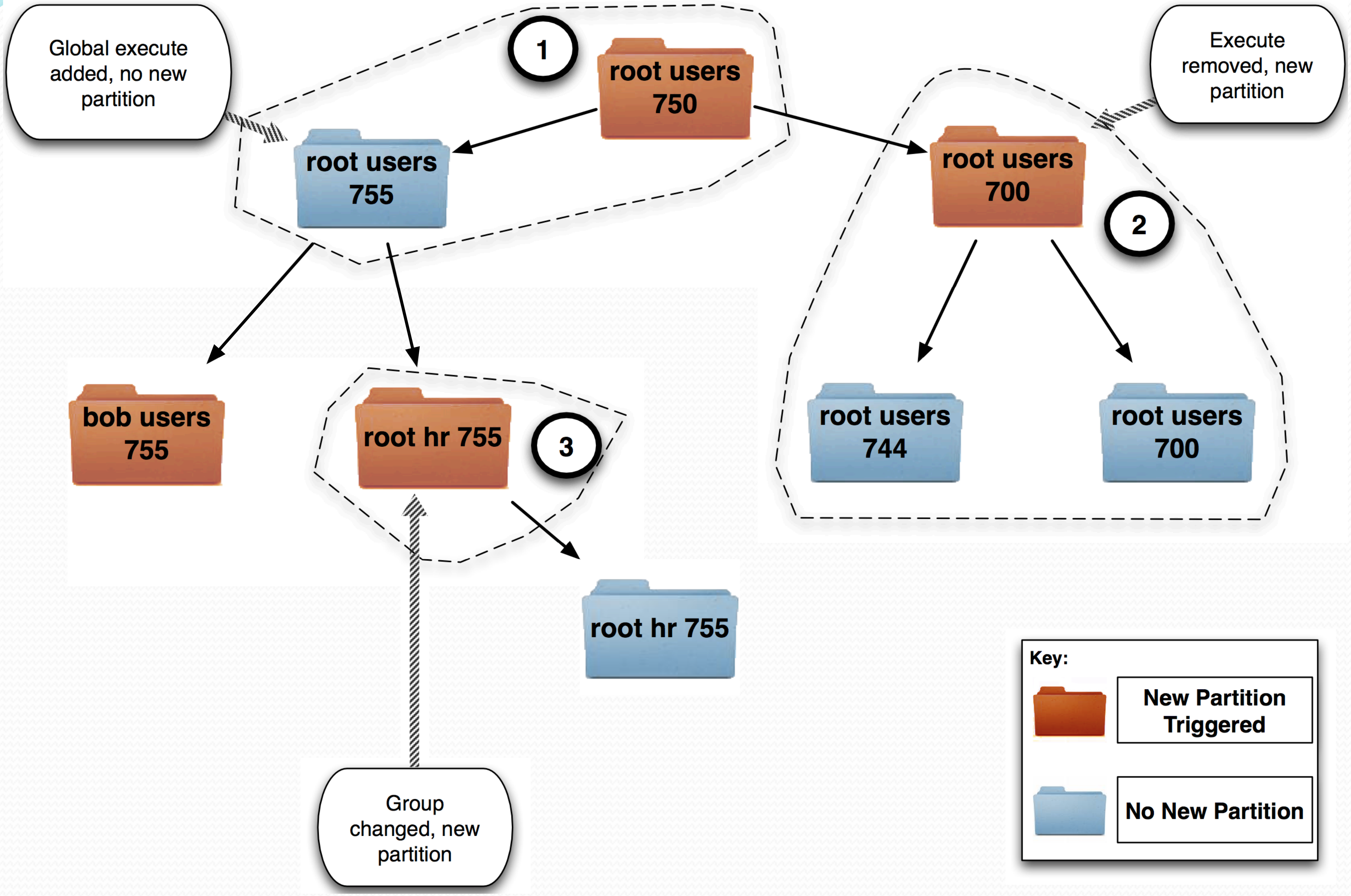
Partitioning Example



Key:

🟧	New Partition Triggered
🟦	No New Partition

Partitioning Example



Global execute added, no new partition

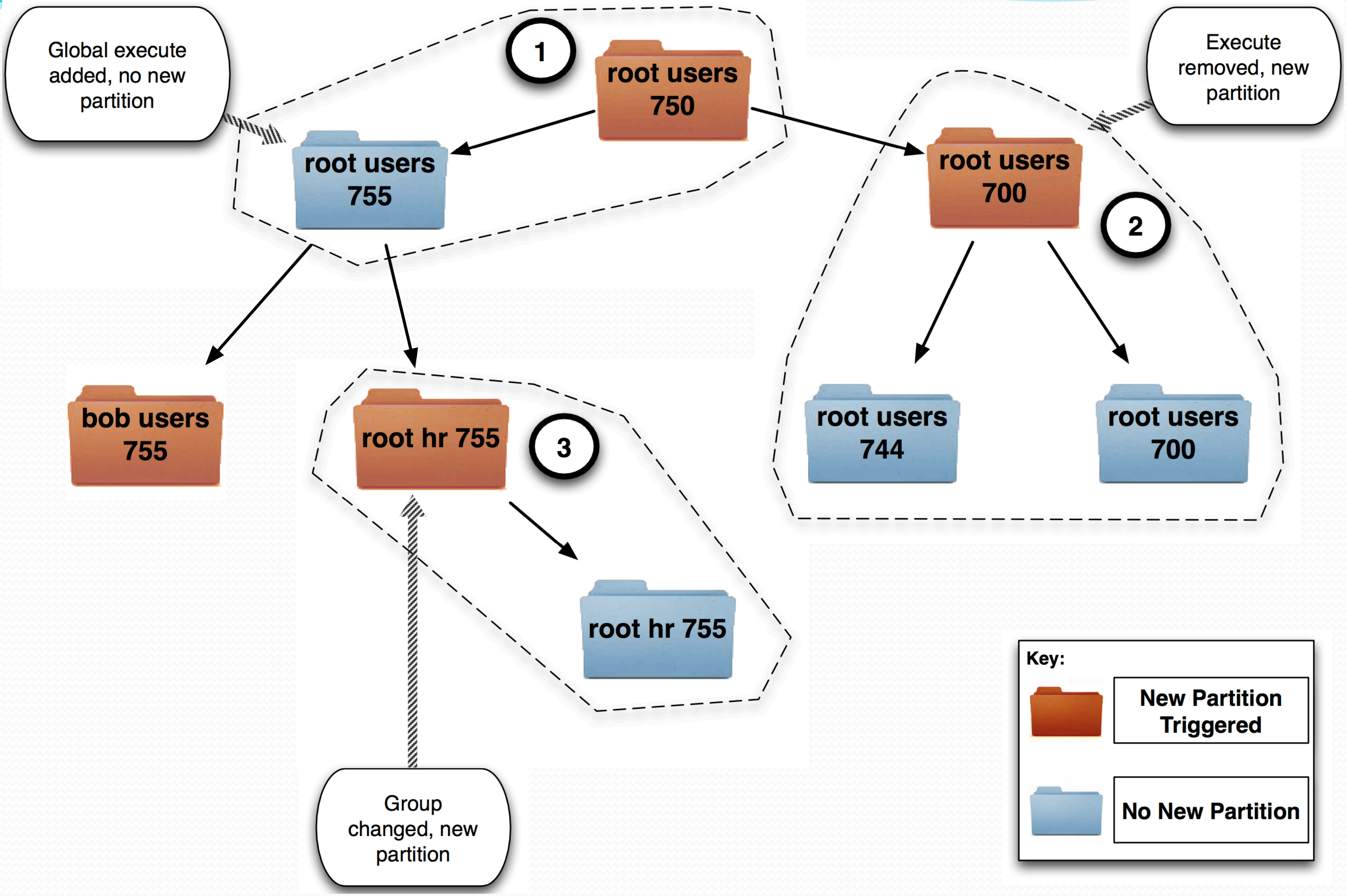
Execute removed, new partition

Group changed, new partition

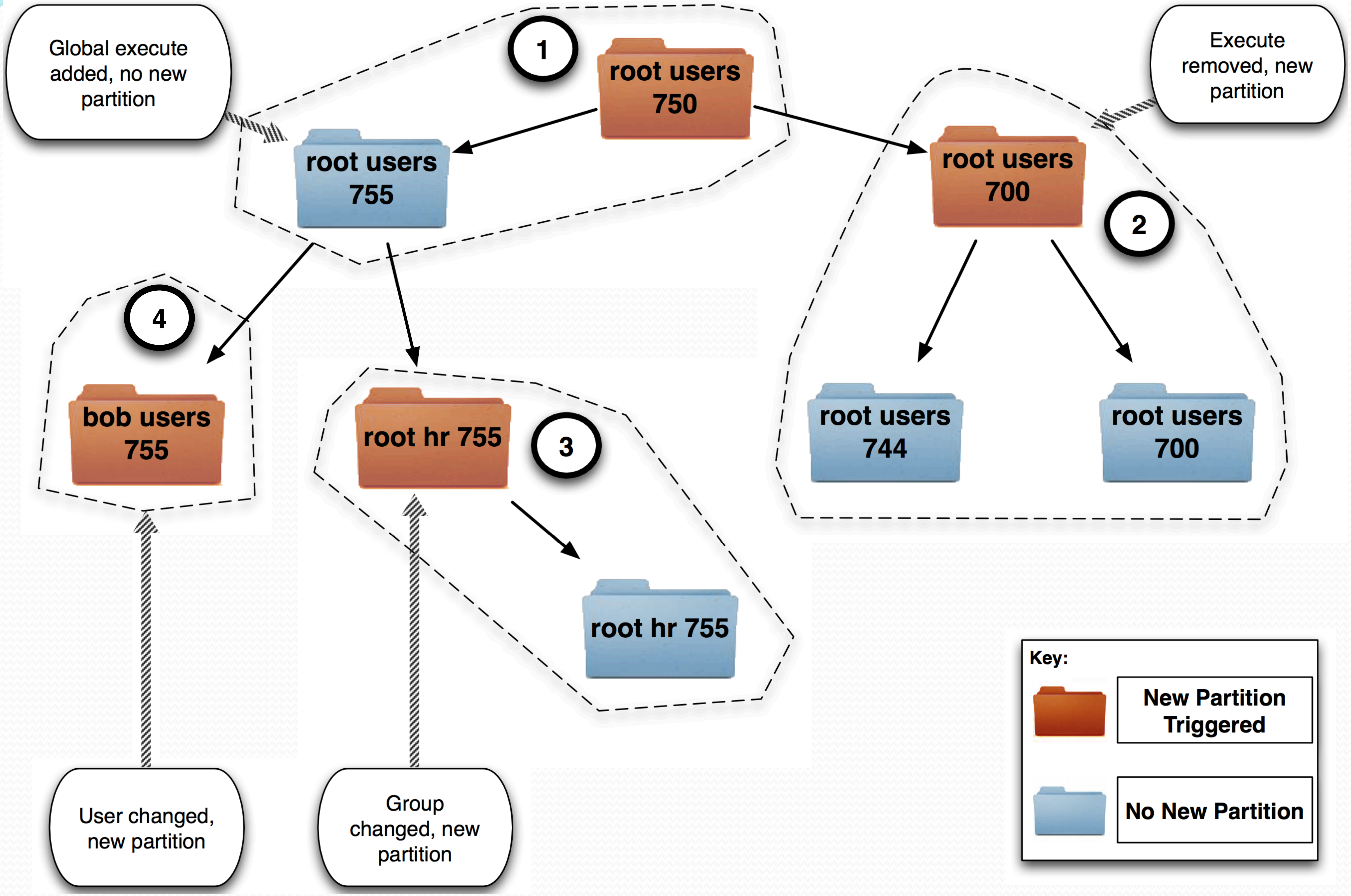
Key:

🟧	New Partition Triggered
🟦	No New Partition

Partitioning Example



Partitioning Example



Global execute added, no new partition

Execute removed, new partition

User changed, new partition

Group changed, new partition

Key:

🟧	New Partition Triggered
🟦	No New Partition

Roadmap

- Security Aware Partitioning Algorithm

★**Metrics for Evaluating Partitioning**

- Evaluation

Testing by Guessing

- Existing algorithms have tested search using ad hoc queries
- Not representative
- Different organizations may have different needs
- Does not fully explore system's capabilities

Evaluation Metrics

- We propose a new set of metrics for evaluating partitioning algorithms
- A complement to benchmarking
- Fully characterizes expected query performance

Metrics

- Run Time
- Memory Requirements
- Partition Size
- Partition Intersection (compares two algorithms)
- Information density
  - Entropy
  - Information Gain

Partition Size

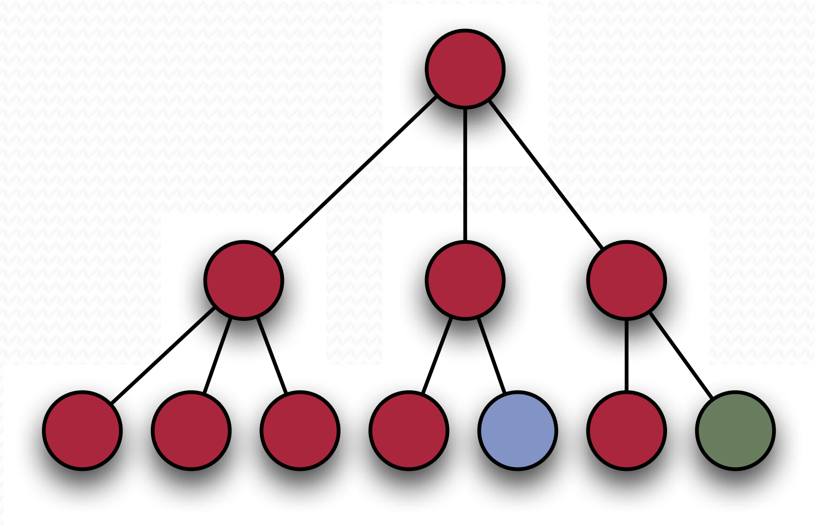
- Tiny partitions are wasteful
  - More operations required to retrieve all matching indexes
- Large partitions can be too big for memory
- Most partitions should be as large as possible for available memory
- We chose a size of 100,000 files, based on prior work (Leung 09)

Partition Intersection

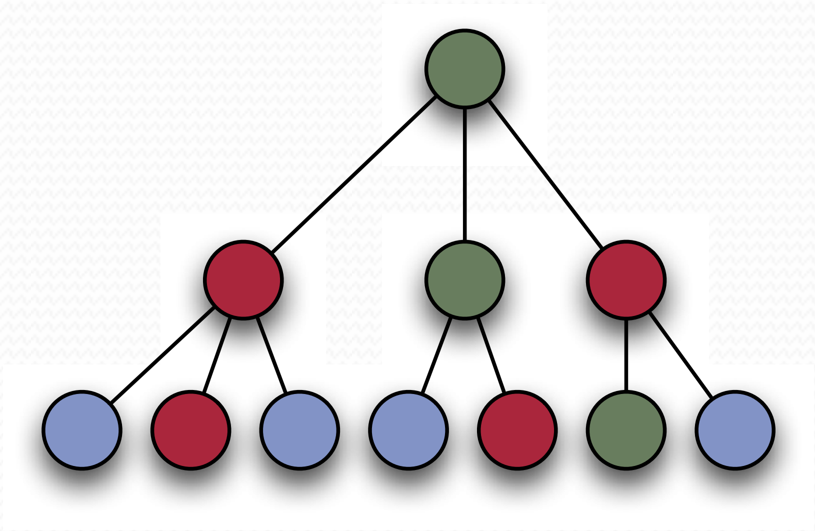
- When comparing two algorithms, do they generate similar partitions?
  - Both algorithms place the same files in the same partitions
- If so, they will behave similarly at query time
- Measure the intersection between partitions
  - Expresses differences in both size and file list
- Comparative, not qualitative

Entropy

- Entropy is a measure of metadata “density” (intra-partition similarity)
- Measures information disorganization
- Low entropy means loading fewer partitions to search for a given value



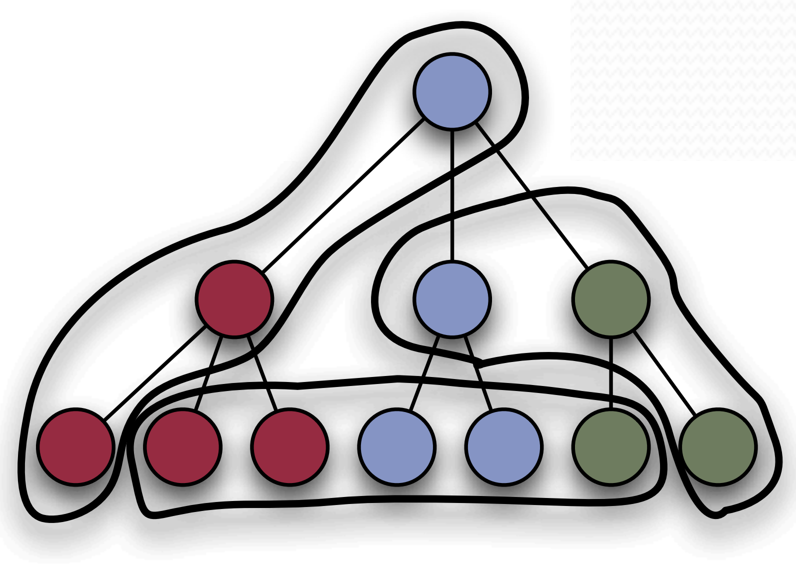
Low entropy



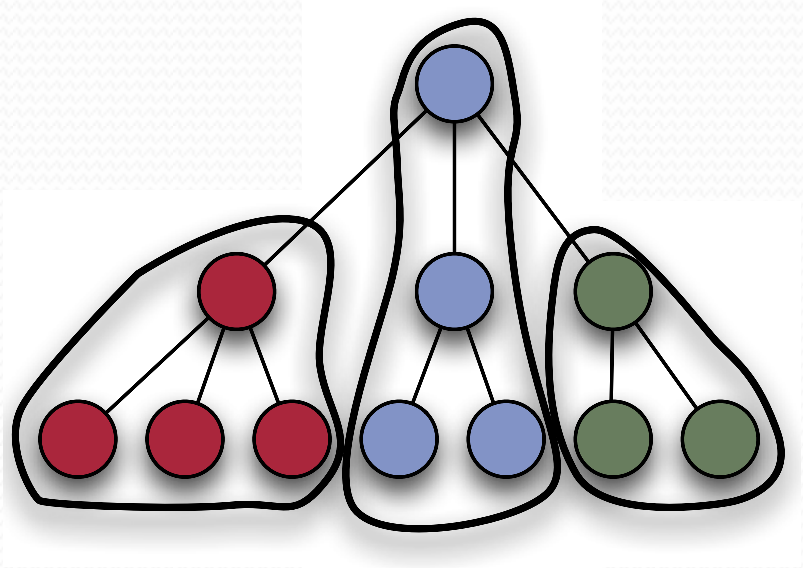
High entropy

Information Gain

- Information gain is a measure of "distinctness" (inter-partition uniqueness)
- How well does the algorithm organize metadata into partitions?
- High information gain means that most files with a specific attribute can be found in a few partitions



Low information gain



High information gain

Roadmap

- Security Aware Partitioning Algorithm
- Metrics for Evaluating Partitioning

★**Evaluation**

Data Sets

- SOE – A crawl of a university file system, shared by undergraduates, graduates, and professors
- Web – NetApp web/wiki server
- Eng – NetApp scratch directory server
- Home - NetApp home directory server

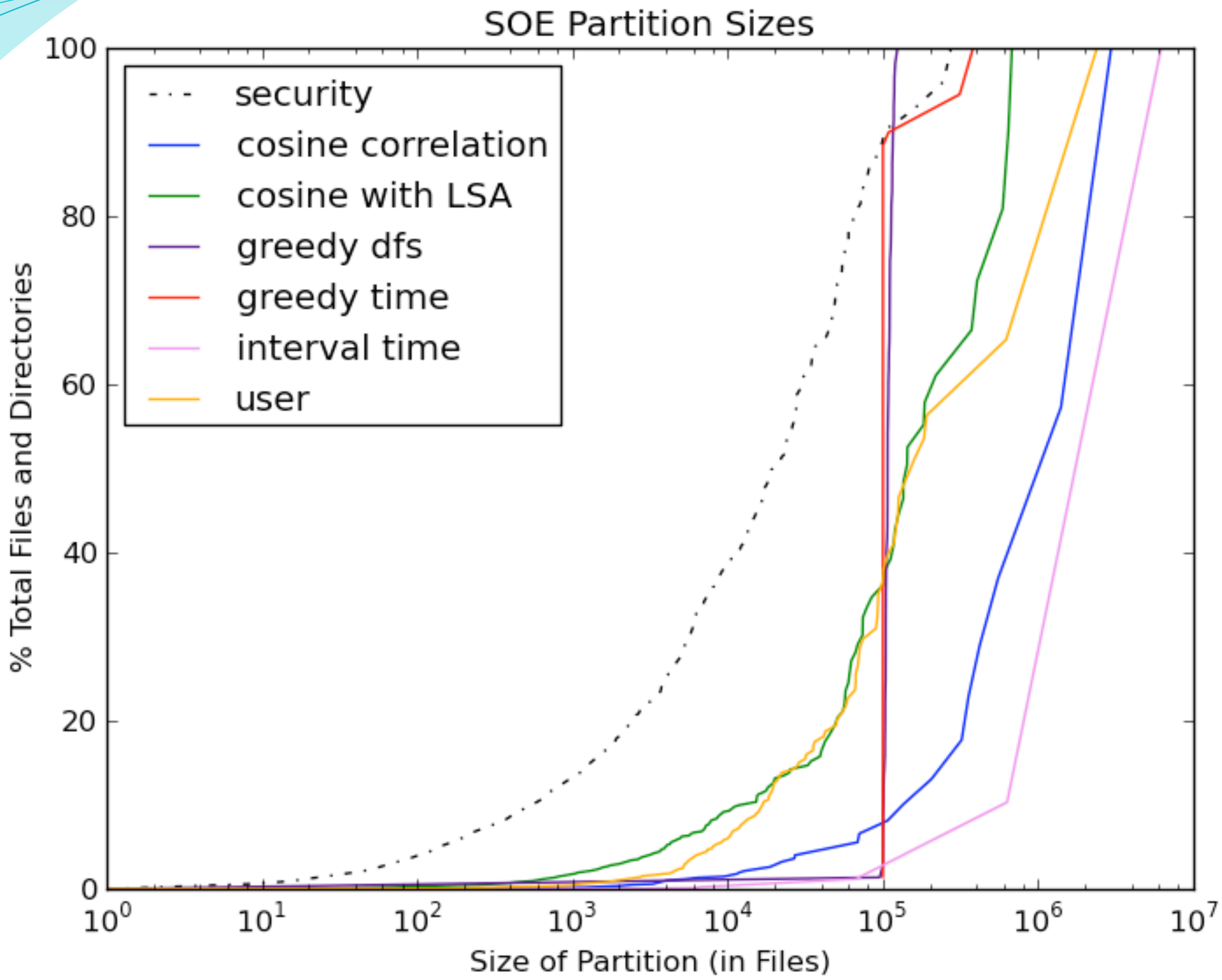
Algorithms Evaluated

- Security Aware Partitioning
- Greedy DFS (Leung 09)
- Cosine Correlation with Latent Semantic Analysis (Hua 09)
- Cosine Correlation (without Latent Semantic Analysis)
- Greedy Time
- Interval Time
- User Partitioning

Run Time and Memory

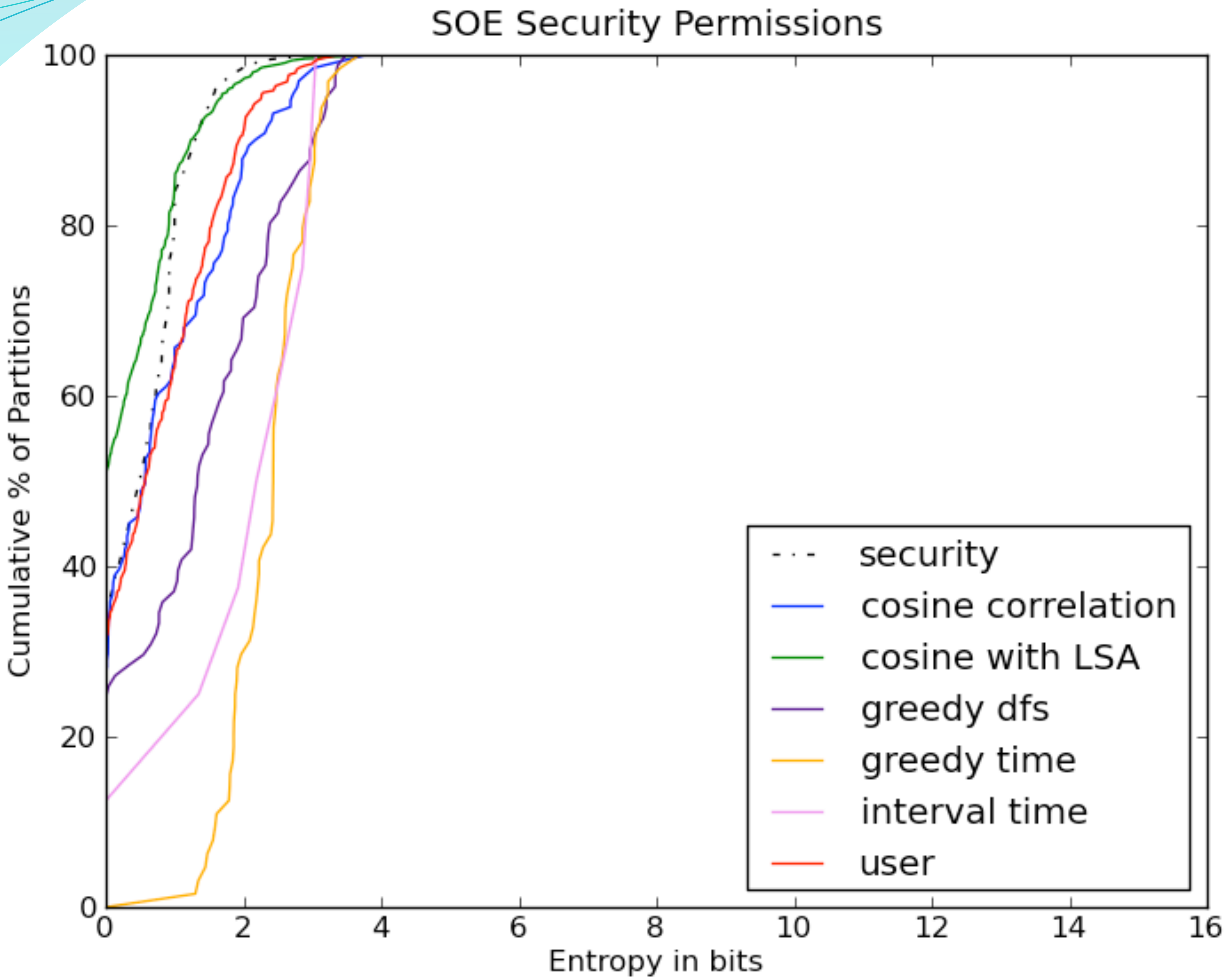
	DFS	Greedy Time	Interval Time	User	Security	Cosine	LSA
Run Time	O(n)	O(n)	O(n)	O(n)	O(n)	O(n^2)	O(n^2) to O(n^3)
Memory	O(1)	O(1)	O(1)	O(1)	O(lg n)	O(n)	O(n)

Partition Size



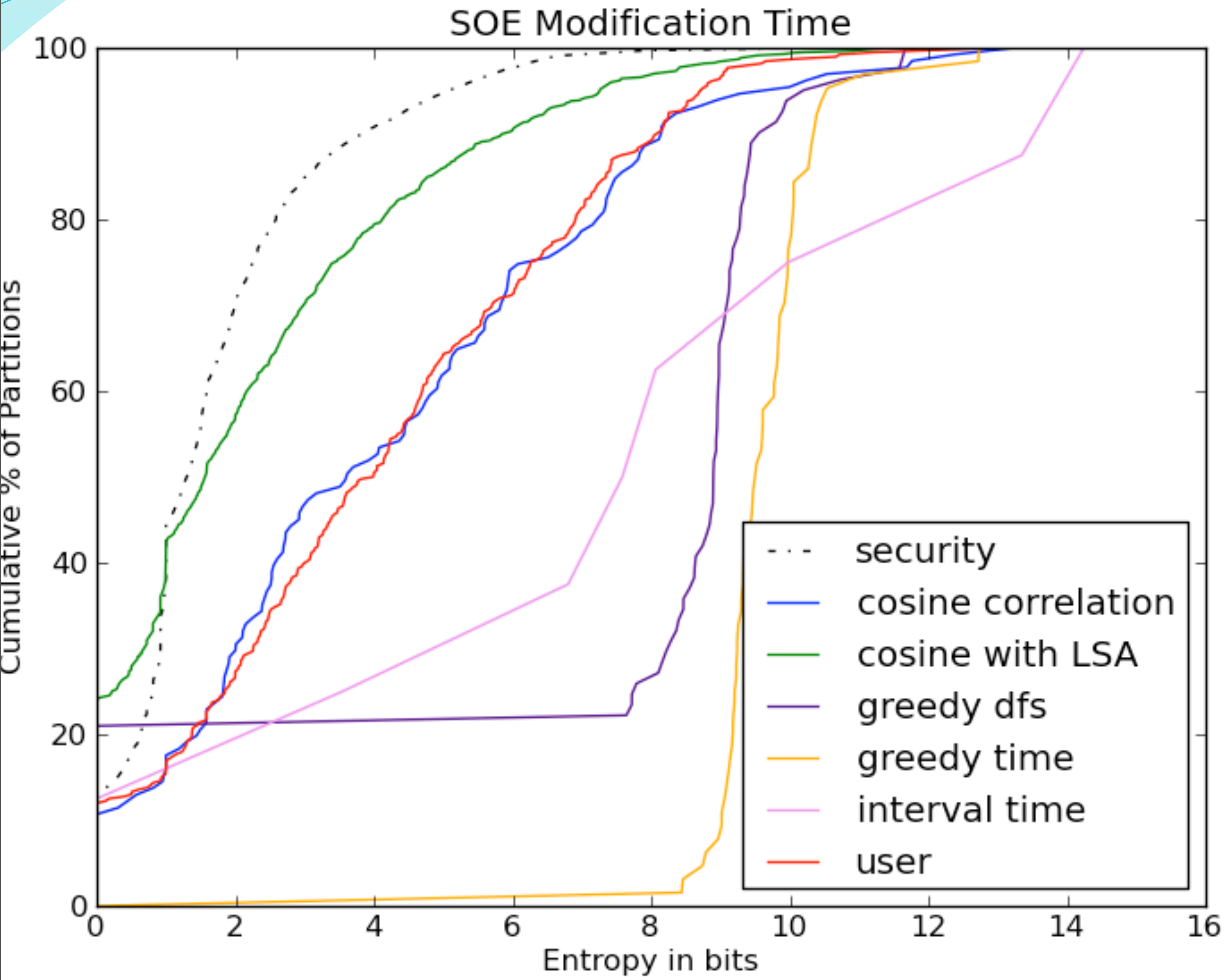
Security Aware Partitioning has smaller partitions

Entropy for Security Permissions



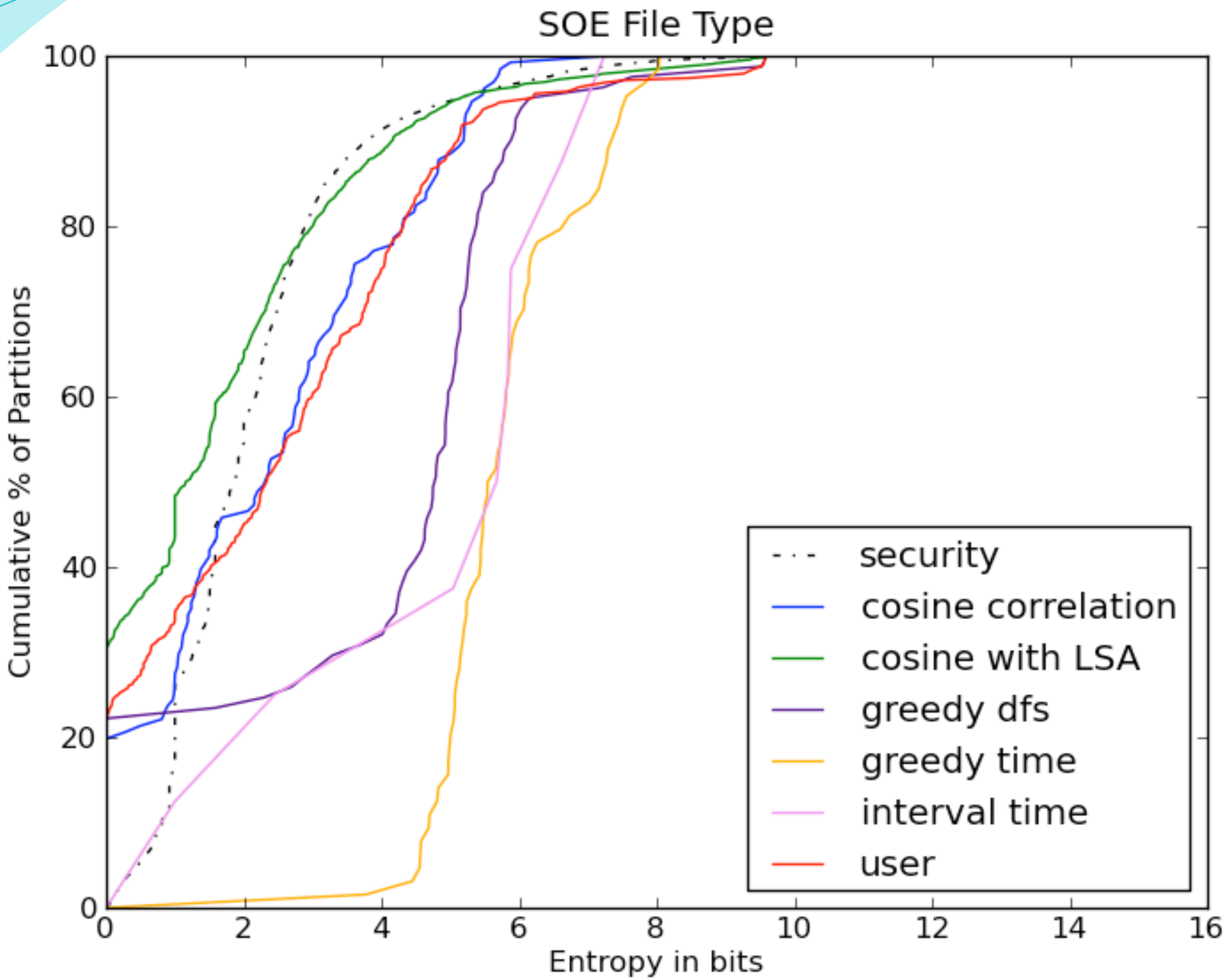
Security Aware Partitioning has low entropy

Entropy for Modification Time



Security Aware Partitioning has low entropy

Entropy for File Type



Security Aware Partitioning has low entropy

Information Gain in Bits

	type	mode	uid	gid	size	atime	mtime	ctime
DFS	3.5	1.5	2.4	1.8	6.6	3.2	7.0	9.0
Greedy Time	7.1	3.0	4.8	3.7	13.2	6.4	14.1	18.2
Interval Time	6.3	2.7	4.2	3.3	11.7	5.7	12.5	16.1
User	3.6	1.5	2.4	1.8	6.7	3.2	7.1	9.1
Security	7.2	3.0	4.8	3.7	13.4	6.5	14.3	14.3
Cosine	7.1	3.0	4.8	3.7	13.2	6.4	14.1	18.1
LSA	7.2	3.0	4.8	3.7	13.4	6.5	14.2	18.4

Summary of Results

Security Aware Partitioning:

- Generates partitions which are, on average, smaller than other algorithms
- Has a run time comparable to greedy algorithms
- Has good entropy and information gain for many attributes
    - Has partition quality comparable to or better than expensive clustering algorithms

Security Benefits of Security Aware Partitioning

- Security Aware Partitioning eliminates filtering operations at query time
- Ensures users can never see results they cannot access
- Prevents statistical attacks on ranked search

Future Work

- Explore richer metadata and content search to further characterize partitioning
- Explore ACLs and how they apply to search
- Implement full prototype in Ceph
- Look at using layout to mitigate cost of smaller partitions

Conclusion

- We have identified metrics that are useful for characterizing a partitioning system without fully implementing it
- We've created a new partitioning algorithm
  - Fast to create
  - Secure
  - Will perform well for many metadata queries

Acknowledgements

- National Science Foundation under awards CCF-0937938 and IIP-0934401 (I/UCRC Center for Research in Intelligent Storage)
- Department of Energy's Petascale Data Storage Institute under award DE-FC02-06ER25768
- Center for Information Technology in the Interest of Society (CITRIS)
- NetApp for making data available to us
- Industrial Sponsors: NetApp, IBM, Symantec, and LSI