Keyword Fusion to Support Efficient Keyword-based Search in Peer-to-Peer File Sharing

Presented by Chi-Hong Chao

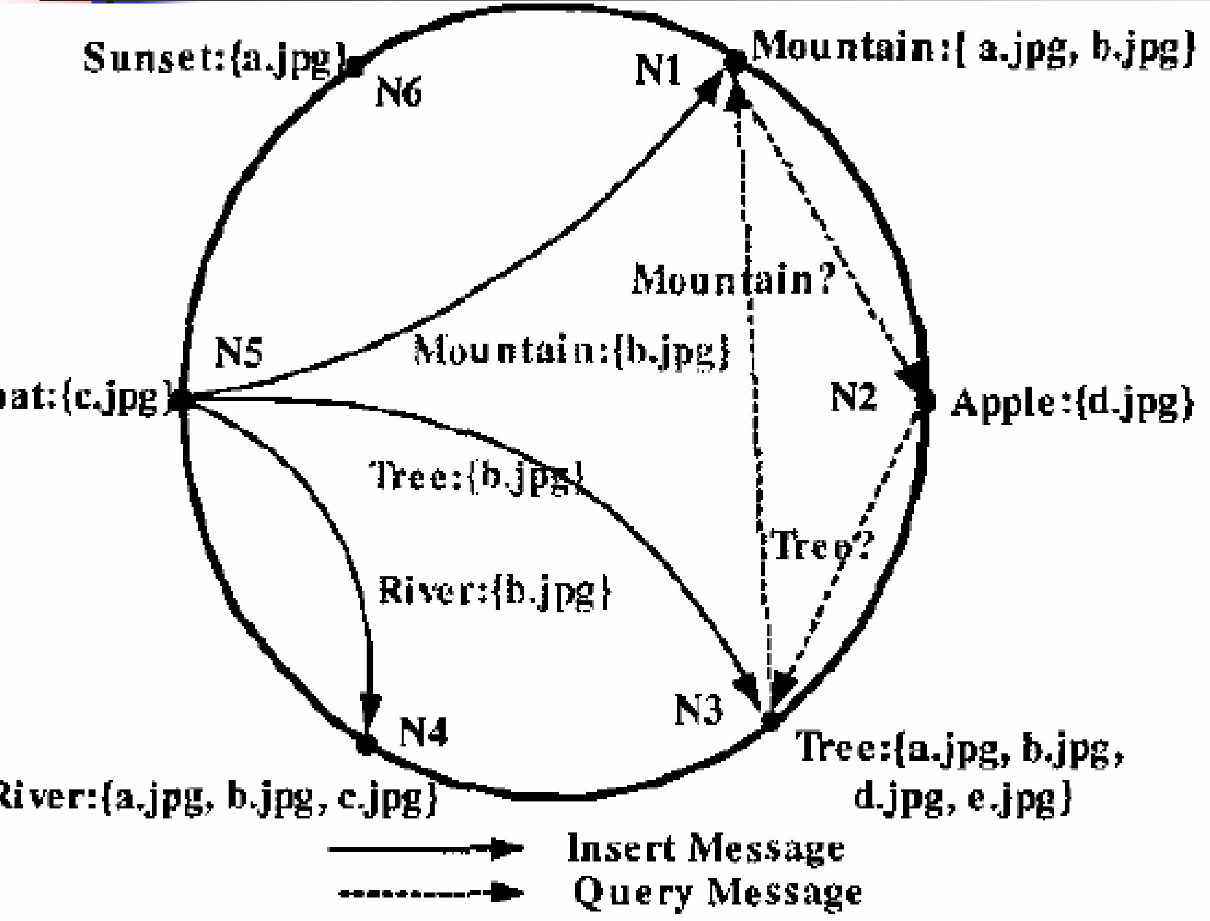
Outline

- Introduction
- Keyword search in DHT-based P2P systems
- Keyword Fusion architecture
- Performance evaluation
- Conclusion and Discussion

Introduction

- Centralized P2P System
  - Napster
- Decentralized P2P System
  - Unstructured P2P System (Gnutella)
    - Flooding
    - Bandwidth consumption
  - Structured P2P System (Chord, CAN,…)
    - DHT
    - Not support keyword search

Keyword search in DHT-based P2P systems – extended Chord



File	Keywords
a.jpg	Tree, River, Mountain, Sunset
b.jpg	Tree, River, Mountain
c.jpg	Boat, River
d.jpg	Apple, Tree
e.jpg	Tree

Keyword search in DHT-based P2P systems –extended Chord

- Chained query processing
  - If N2 wants to find files containing both "**Tree**" and "**Mountain**", N2 can send out a query message to N3 which is responsible for **Tree**.
  - N3 then sends the intermediate result set {a.jpg b.jpg d.jpg e.jpg} to N1, where the file list of **Mountain** is stored.
  - By intersecting the intermediate results from N3 with the file list for **Mountain**, N1 will generate the final result, {a.jpg b.jpg}

Keyword search in DHT-based P2P systems

- The chained query processing is conceptually simple, but there is a few drawback. (Commonly keywords)
  - Storage consumption is highly skewed among peers.
  - The common keyword will generate huge volume of network traffic.

Keyword Fusion architecture
- Preliminaries

- Definition
  - h(k): the hosting DHT node which stores the mapping for keyword k.
  - K(f): the set of keywords associated with file f.
  - F(k): the set of files which contains keyword k.
- Keywords in the query are AND-ed.

Keyword Fusion architecture
- Fusion Dictionary

- The common keywords are the cause of the two main problems (storage and network traffic).
- When searching for files that contain multiple keywords, it’s advantageous to search for the **most specific keyword** first.

Keyword Fusion architecture
- Fusion Dictionary

- Fusion Dictionary is a distributed data structure that contains common keywords.
- When a DHT node determines that is storage consumption is excessive, it identifies common keywords from its list and registers them to Fusion Dictionary.
- After this registration it removes the mapping information for the common keyword from its storage.

Keyword Fusion architecture
- partial keyword list

- As the mappings for common keywords are removed from the hosting nodes, it's required to have a mechanism to handle queries containing such deleted keywords.
- For each file *f* in the mapping $<k, \mathrm{F}(k)>$, we create and store a partial keyword set $\mathrm{PK}(f)=\mathrm{K}(f) \cap \mathrm{FD}$.

Keyword Fusion architecture
- Fusion Dictionary & partial keyword list

- When a node issues a search query, it first consults Fusion Dictionary to select the keywords which are not in the dictionary, then access their hosting nodes in a chain for query processing.
- With partial keyword lists added to file list, the common keywords in the query can be processed at any of those nodes.
- This will make query processing more efficient by omitting the nodes hosting common keywords and avoiding transferring large intermediate results.

Keyword Fusion architecture
- Fusion Dictionary & partial keyword list

- Since the partial keyword list PK(f) is determined by the current Fusion Dictionary, it's also generated and maintained dynamically.
- When a keyword k is added into Fusion Dictionary, the node h(k) just removes all the entries in F(k) and propagates the dictionary update to other nodes.

Keyword Fusion architecture
- Fusion Dictionary & partial keyword list

- When a node receives a dictionary update that k is added into the Fusion Dictionary, it first checks its local database.
- If this node has published a file f which contains k as one of its keywords, it re-publishes the same file f into the network by sending it to the nodes hosting keywords other than k in K(f).

Keyword Fusion architecture
- Fusion Dictionary & partial keyword list

- In order to minimize the lookup overhead, the content of Fusion Dictionary is replicated and propagated across DHT nodes.
- Managing Fusion Dictionary and partial keyword lists is a fully decentralized operation.
- The Fusion Dictionary updates can be included in these periodically topology maintaining messages.
- The query messages can also be used to piggyback the dictionary updates. Since these query messages are traveling around the whole network, it will greatly accelerate the propagation progress.

Keyword Fusion architecture
- Keyword Fusion

- There is a file that associated with multiple keywords a and b, we can safely remove this file's information from node h(a) as long as the entry for keyword b is maintained because the file is still searchable using the remaing keyword b.
- Now what happens when h(b) decides that keyword b is generic and must be removed from its hosting DHT node? Such situations are handled by Keyword Fusion.

Keyword Fusion architecture
- Keyword Fusion

- Combine
  - $K=\{K_1,K_2,\ldots,K_n\}$
  - Combine(K)=K'="K_1&K_2&…K_n"
  - Example:
    - Combine(Tree,River) generate a new keyword "Tree&River".
  - We call the new keywords to be ***synthetic keywords*** to distinguish them from the ***original keywords***.

Keyword Fusion architecture
- Keyword Fusion

- Assume Fusion Dictionary contains keywords, $\mathbf{a_1,a_2,\ldots,a_m}$. Now suppose a keyword **b** is added into the Fusion Dictionary from its hosting node h(b).
- New keywords are generated by combining b with all the keywords in the Fusion Dictionary and the new synthetic keywords are inserted into P2P network using consistent hashing along with their mapping.

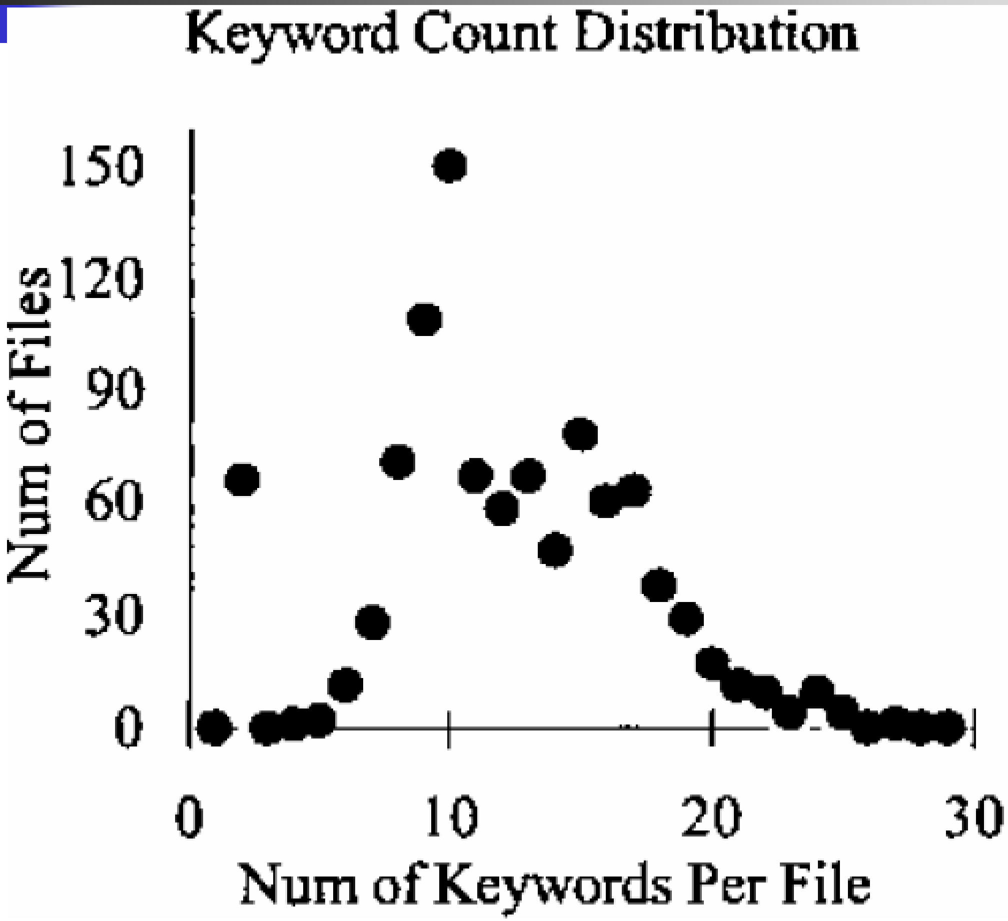
Keyword Fusion architecture
- Keyword Fusion

- More precisely, Keyword Fusion ensures that all the keywords in the Fusion Dictionary that are combined in a pair-wise manner do exist in DHT.
- Example:
  - Fusion Dictionary={a,b,c}
  - Keyword Fusion guarantees that synthetic keywords **a&b**, **b&c**, **a&c** exist in the DHT.

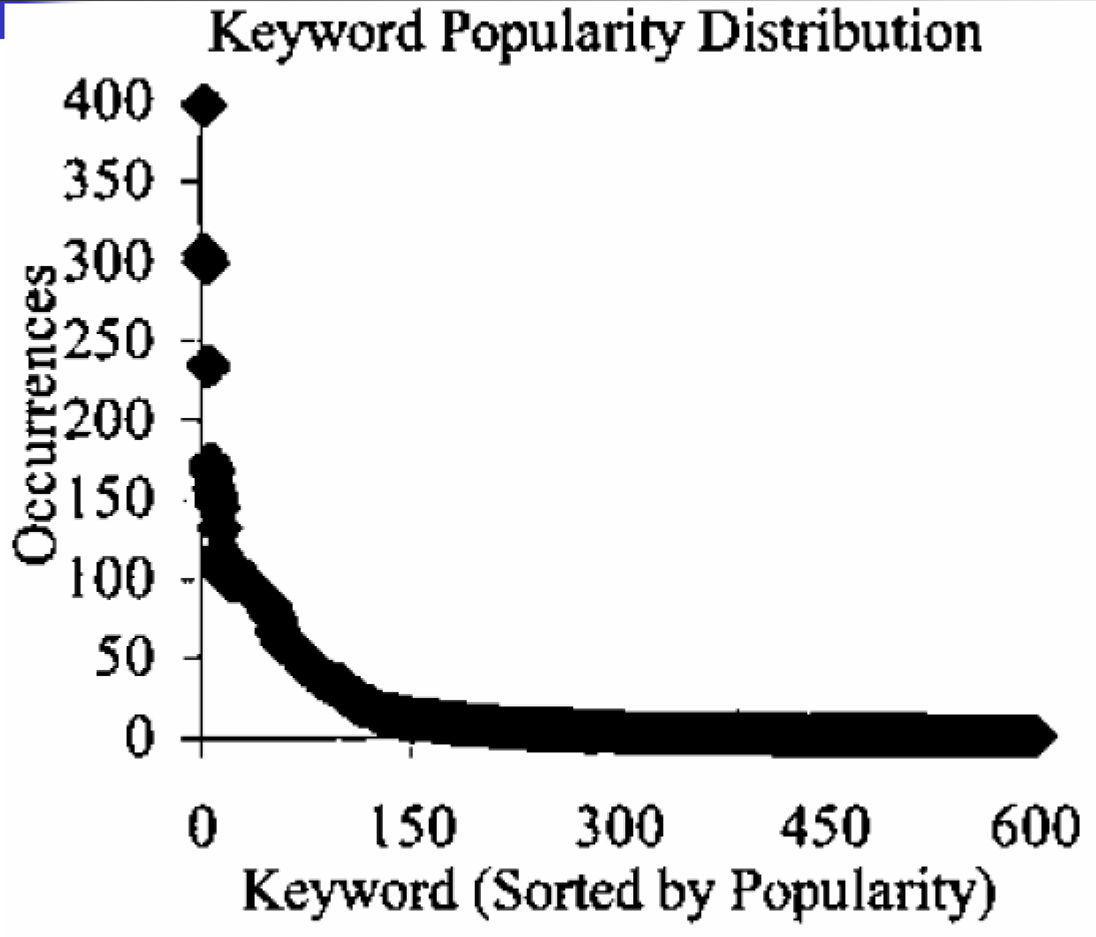
Performance Evaluation

- Data set A:
  - 1,000 images annotated with relevant keywords
- Data set B:
  - 40,000 images
  - More than 38,000 of these files are annotated with 4 keywords selected from 6,510 unique keywords.

Performance Evaluation
(data set A)

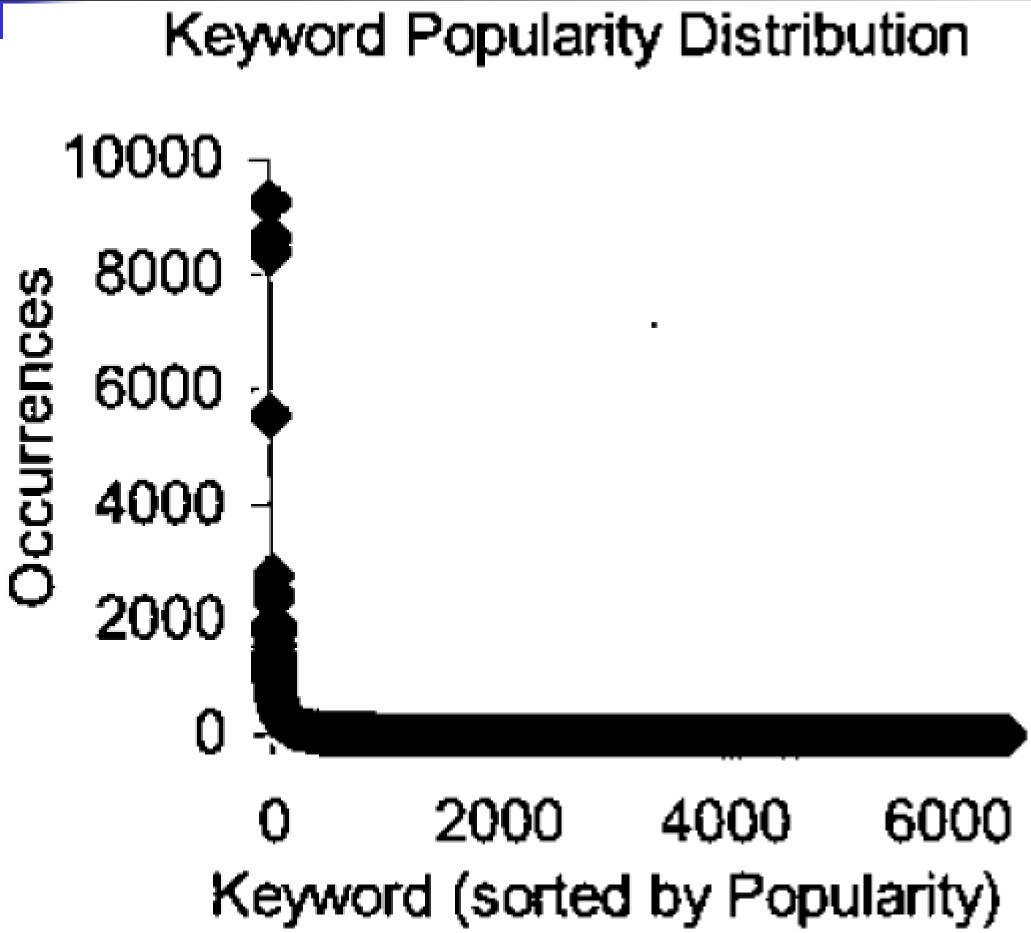


Performance Evaluation
(data set a)



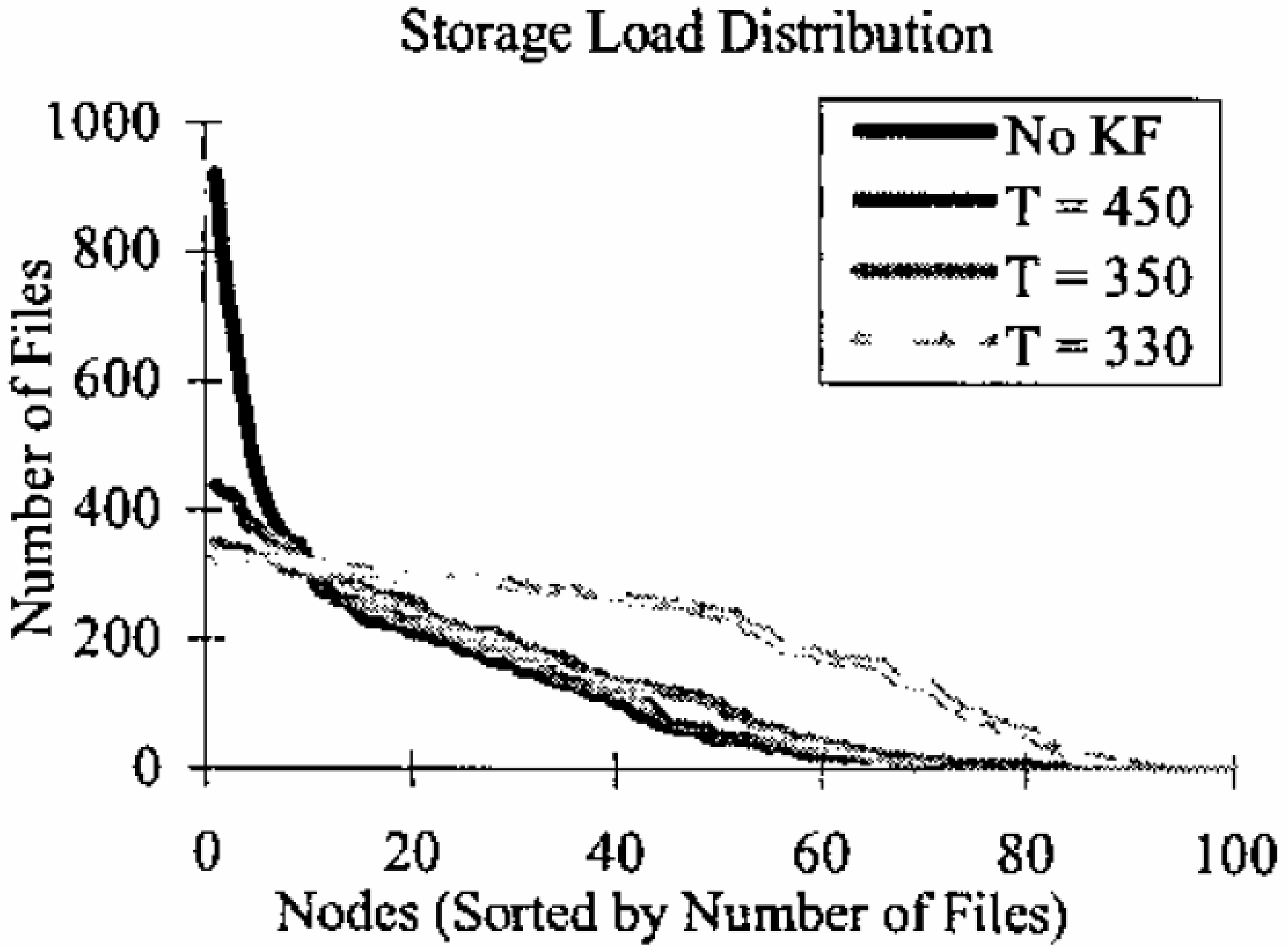
- Top 5% most frequent keywords appear 6,608 times.

Performance Evaluation
(data set b)

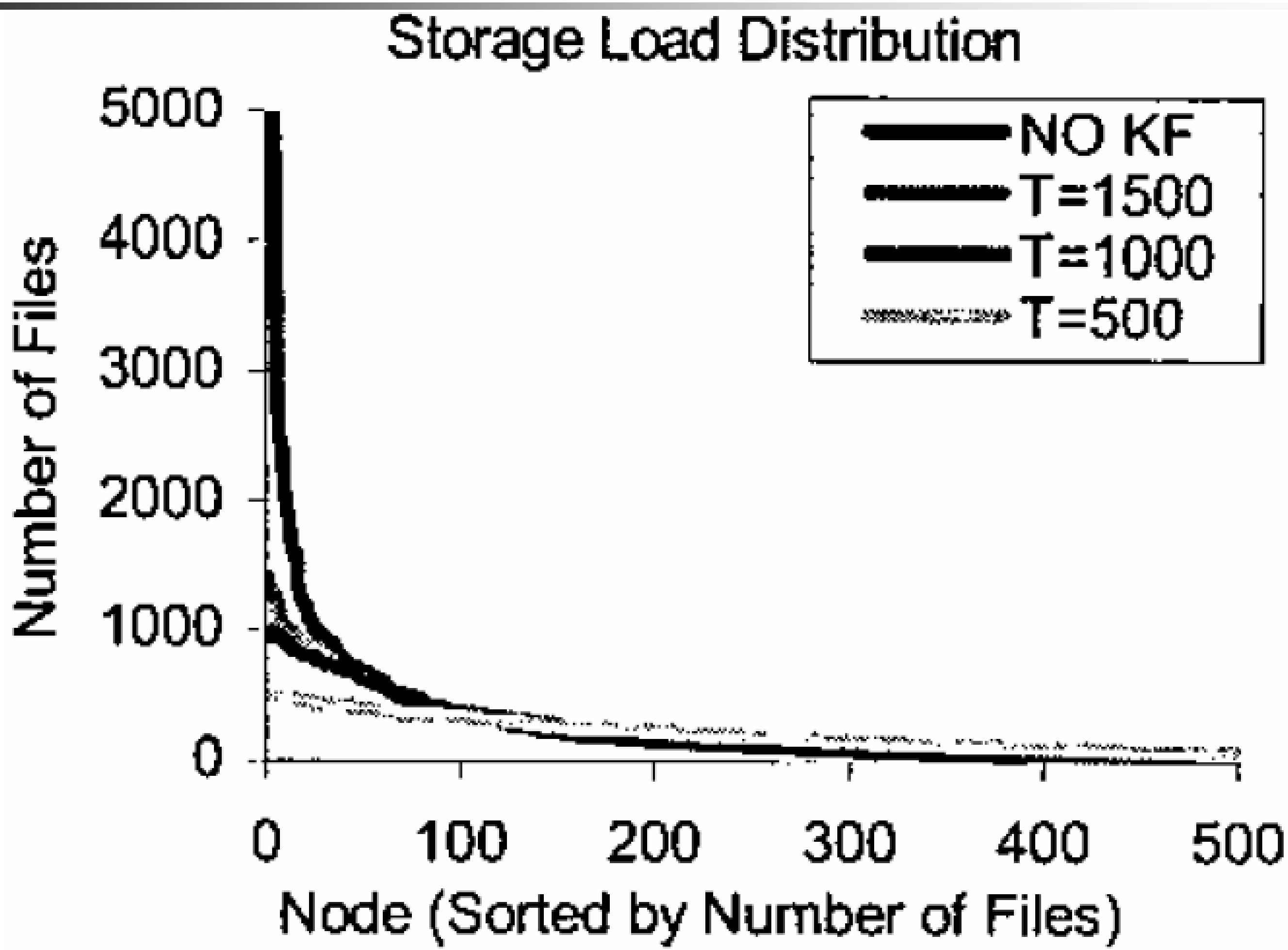


- Top 5% most frequent keywords appear 124,534 times, out of the total 161,051 keyword occurrence .

Performance Evaluation
(data set a)



Performance Evaluation
(data set b)



Conclusion and Discussion

- Keyword Fusion can reduce the search traffic by up to 68%.
- Keyword Fusion also can effectively unburden overloaded peers and distribute the file storage load across the entire DHT network.
- But how to analyze a file into multiple keywords is still a big problem now!