

Popularity-aware Prefetch in P2P Range Caching

**Presenter: Qiang Wang
University of Waterloo**

**Joint work with Khuzaima Daudjee
M. Tamer Özsu**

P2P Computing 2008, Aachen

Outline

- ❖ **Motivation**
- ❖ **Problem Analysis**
- ❖ **Correlation Measurement**
- ❖ **Prefetch Mechanism**
- ❖ **Simulation Results**
- ❖ **Conclusion**

Motivation

❖ P2P is an infrastructure for widely distributed applications



Motivation

❖ Range query processing is important for many applications

- Multimedia content retrieval (temporal)
- Location-based services (spatio)

❖ For example

- Search all songs of rock band U2 *between year 1987 and 1997*
- Find the hotels that are *within two miles from the conference venue*

Problem Analysis

- ❖ **Existing search approaches in unstructured P2P networks employ random-walk or gossip mechanisms [1]**
- ❖ **The search cost (e.g., the number of messages) is inversely proportional to the number of replicas of the data results in the network**

Problem Analysis

- ❖ **Since a range query q may include multiple data items $\{s_1, s_2, \dots\}$. Some data items are well-replicated, while others are poorly-replicated**
- ❖ **The query processing cost is decided by those data items that are poorly-replicated in the network**

$$\text{cost}(q) \propto \frac{1}{r_q}, \text{ where } r_q = \{|s_1|, |s_2|, \dots\}$$

Problem Analysis

❖ **Under existing replication mechanisms, range query processing (including poorly-replicated data) may get affected**

- Proportional replication scheme (in Gnutella [2])

$$\text{cost}(q) = \sum_{i=1}^m \frac{N}{i} \rightarrow N \times \ln m$$

- Square-root replication scheme [1]

$$\text{cost}(q) = \sum_{i=1}^m \frac{N}{\sqrt{i}} \quad \text{the cost does not converge when } m \text{ is very large}$$

Problem Analysis

❖ **The goal is to facilitate the replication of those poorly replicated data that are potentially queried in future range queries**

Design tasks

- Locate those poorly-replicated data that will be queried in the future
- Replicate these data to improve query processing performance

Problem Analysis

❖ **To locate the poorly-replicated data**

- Insight: popular data can be measured by their number of replicas (i.e., widely-replicated) in the network; those poorly-replicated data that are *correlated* to popular data are potentially queried in the near future
- Thus, we need to find those poorly-replicated data that are correlated to the widely-replicated data in the system

Problem Analysis

❖ **To replicate the poorly-replicated data**

- Insight: prefetch mechanism can achieve this goal
 - for instruction feed of operating system [3]
 - for online video system [4]
- Thus, we adopt prefetch mechanism and allow the query processing over well-replicated data to piggyback the correlated poorly-replicated data

Correlation Measurement

- ❖ **Range span of normal queries is domain-specific**
- ❖ **Correlation is measured by Euclidean distance D between point data items, or between centroids of range data items**

- Two data items s and s' are correlated if

$$D(s, s') \leq \tau, \text{ where } \tau \text{ is correlation threshold}$$

Correlation Measurement

❖ **Choosing correlation threshold by manual is not sufficiently adaptive**

❖ **Our approach**

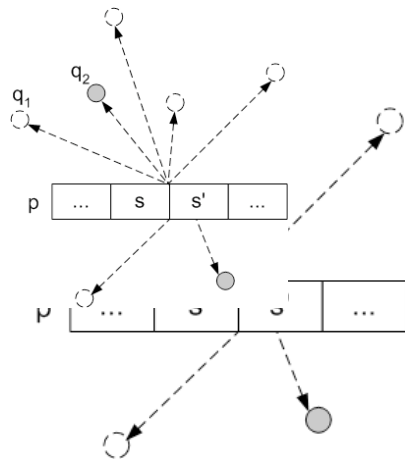
- Each peer collects query histories from other peers. Query history includes the distances between well-replicated data item and poorly-replicated data items that are covered by previous queries
- The expected distance between well- and poorly-replicated data items is computed via approximation (non-parametric kernel estimation or simply mean)

Correlation Measurement

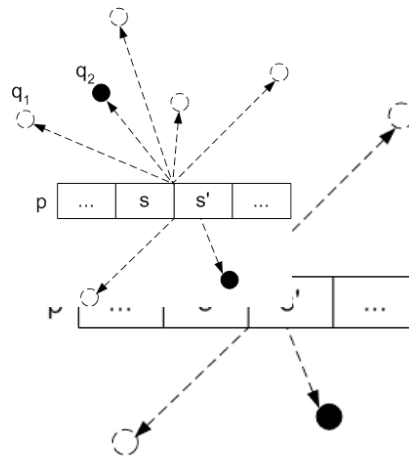
- ❖ **Prefetching threshold is decided by the query issuer. Thus query issuers can consider other criteria (e.g., local storage capacity)**
- ❖ **The popularity (whether well- or poorly-replicated) of data items is measured independently by each peer via sampling-based exploration [5]**

Prefetch Mechanism

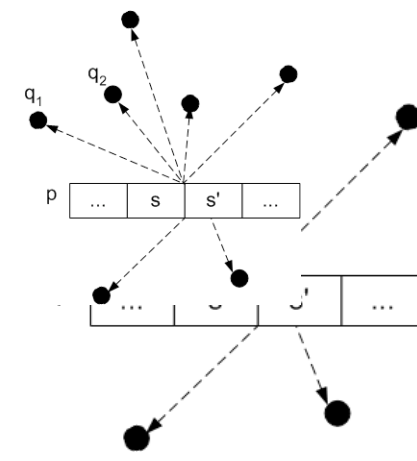
- ❖ Regarding query processing, those peers that hold well-replicated data results *piggyback* the correlated poorly-replicated data with the results



without replication



proportional replication



with prefetching

Prefetch Mechanism

- ❖ Under certain query processing load over well-replicated data items, the performance of query processing over correlated poorly-replicated data items is guaranteed to improve
- ❖ Denote by $\frac{m_1}{m_2}$ the ratio of the number of queries over well-replicated data against that of the queries over correlated poorly-replicated queries

When $\frac{m_1}{m_2 n} \geq 2^\delta$, the overall cost for processing m_2 queries

including poorly - replicated data equals to $O(1) \times N$, where $\delta > 1$.

Prefetch Mechanism

- ❖ **Performance improvement: for example, when $\delta = 1.5$, $\text{cost}(q_2) = 2.612 \times N$, instead of being proportional to $\ln m_2$**
- ❖ **It is also proved that, when network churn and cache expiration happen, under certain query load, same improvement can still be achieved**

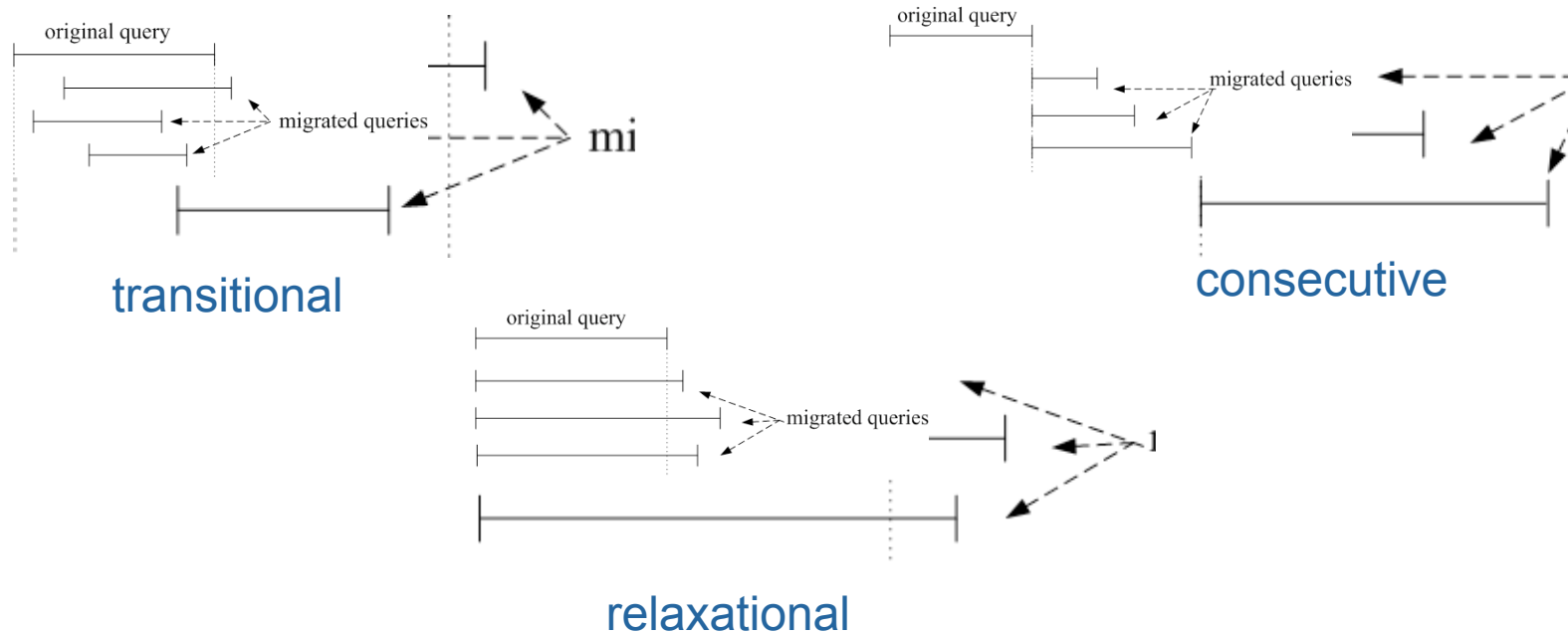
Prefetch Mechanism

- ❖ **Popularity-awareness: only poorly-replicated data are prefetched, although the prefetching of well-replicated data items may also facilitate query processing, it is not cost effective because well-replicated data items are inherently cheap to retrieve**

Simulation Results

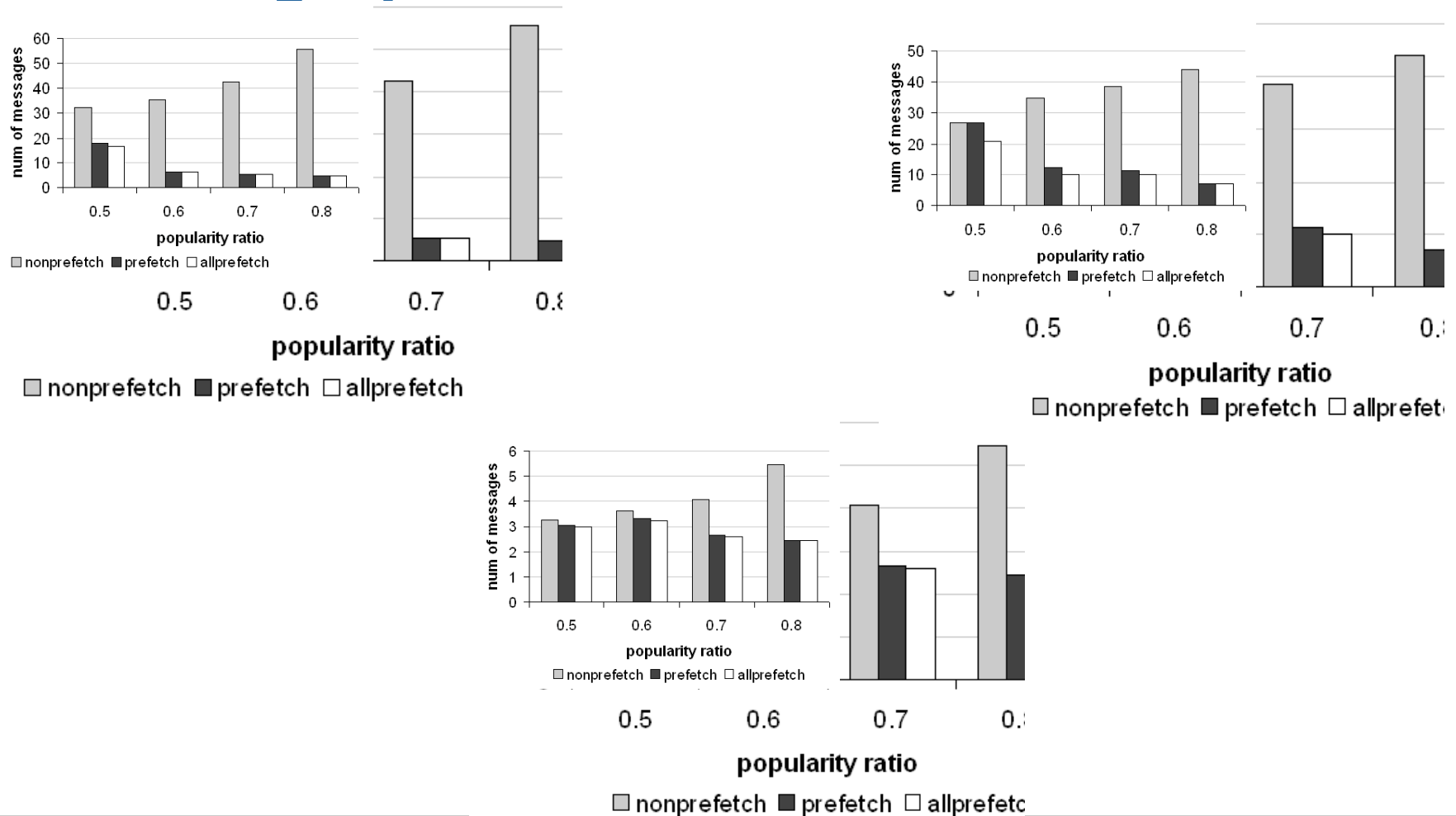
❖ Simulation setting

- p2psim [6] simulating unstructured P2P network with 3000 peers
- Synthesized one-dimensional range queries
- Query migration



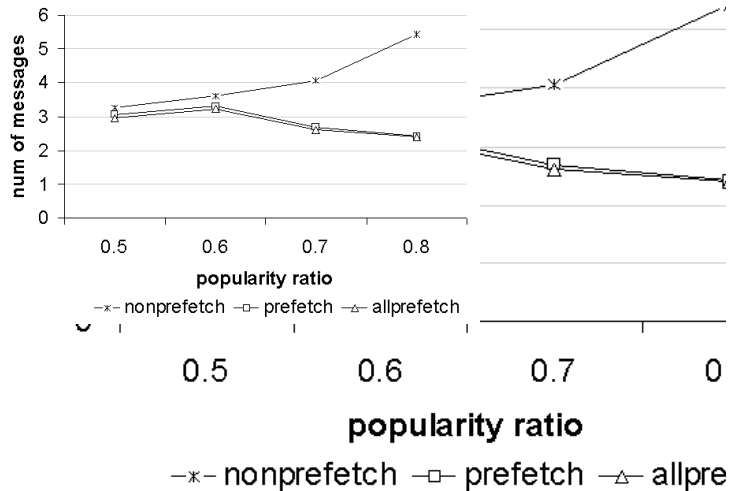
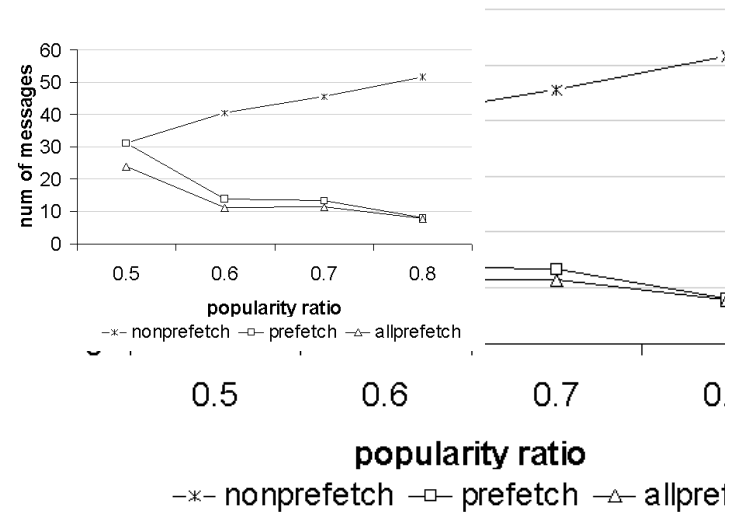
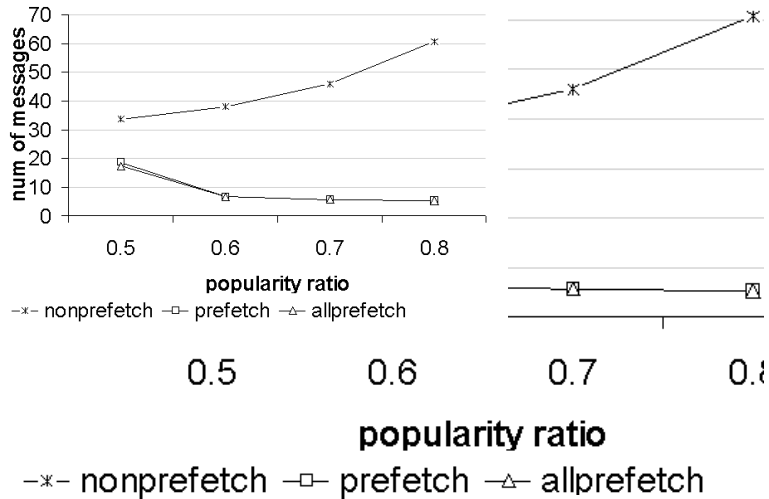
Simulation Results

❖ Overall query processing cost (number of messages)



Simulation Results

❖ Processing cost of migrated queries



Conclusion

- ❖ **In unstructured P2P networks, when range queries involve poorly-replicated data items, the performance will get impacted**
- ❖ **We propose a prefetch mechanism to facilitate overall query processing, which is popularity-aware and adaptive**
- ❖ **We prove that the overall query processing performance will improve under certain query distribution; we also demonstrate the performance improvement with simulations**

References

- ❖ **[1]** E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. In Proc. ACM SIGCOMM, pages 177-190, 2002.
- ❖ **[2]** Gnutella: <http://www.gnutella.com>
- ❖ **[3]** W. Stallings. Operating Systems: Internals and Design Principles. Prentice Hall, 2004.
- ❖ **[4]** B. Cheng, X. Liu, Z. Zhang, and H. Jin. A measurement study of a peer-to-peer video-on-demand system. In Peer-to-Peer Systems, First International Workshop, 2007.
- ❖ **[5]** R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica. Querying the internet with PIER. In Proc. 29th Int. Conf. on Very Large Data Bases, pages 321-332, 2003.
- ❖ **[6]** p2psim: <http://pdos.csail.mit.edu/p2psim/>

Thank You