XPath Lookup Queries in P2P networks

Angela Bonifati, *Icar CNR*, Italy

Joint work with: Ugo Matrangolo and Alfredo Cuzzocrea

(*University of Calabria, Italy*), Mayank Jain* (*IIT, New Delhi*)

*Work done while the author was visiting Icar CNR as a summer intern

Outline

- Keywords: P2P, XPath, XML
- “Store and Query” XML over peers:
  - Motivations
  - Data Model
  - Paths as identifiers
  - Query Answering Algorithms
  - Experimental Study
    - Our prototype: XP2P (Xpath for P2P)
  - Summary and Future Work

XML over P2P: what is out there

- P2P networks advertise keyword-based queries
- XPath is more than that:
  - Q: Document → Set of Nodes
- Useful to understand how XML documents might be stored on peers
- Relevant body of research:
  - interesting results for small communities of XML peers (Pitoura04, Galanis2003)
  - scalable solutions for range queries for relational data (ElAbbadi2003, Gehrke2004)

Locating peers

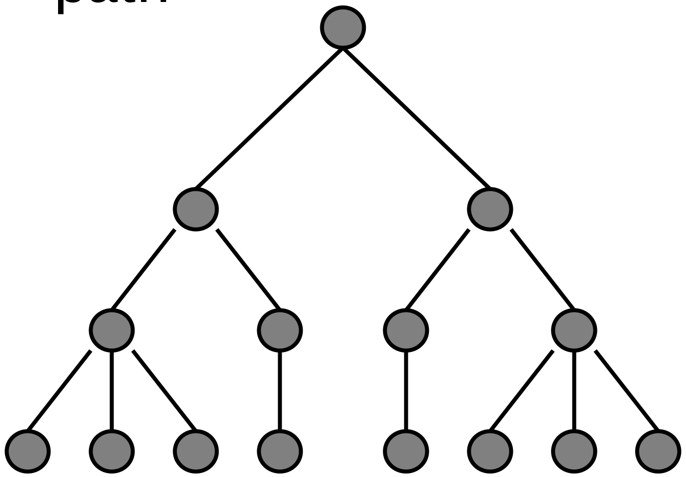
- Actually existing solutions base on:
  - Replicating locally complex structures:
    - Path summaries
    - Bloom-filters, B+-trees, Histograms
  - Drastically limit numbers:
    - Depth of documents
    - Number of peers
- Our solution:
    - Use XPath itself to locate peers
    - Store locally as less *global data* as possible

XP2P:The problem

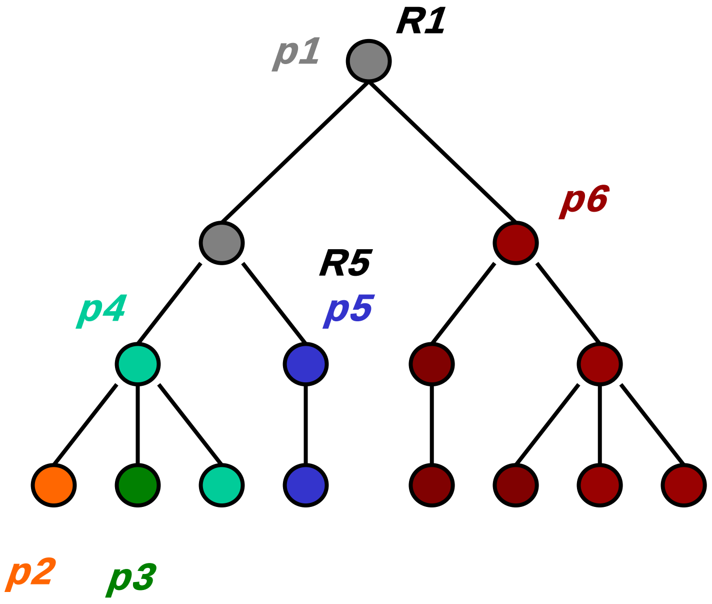
- What XP2P is:
  - A framework for storing *XML fragments*
  - and for answering XPath in a *structured* network
- What XP2P is not:
  - A framework for handling a large number of *replicas*
  - and for answering XPath in an *unstructured* network
  - or in *hybrid* networks (gnutella+PierDB, Loo2004)

What do we mean by XML fragments?

- A subdocument of the original document
  - given a document D, a *fragment* thereof is defined as a *subtree* of the original document and identified by its absolute linear path



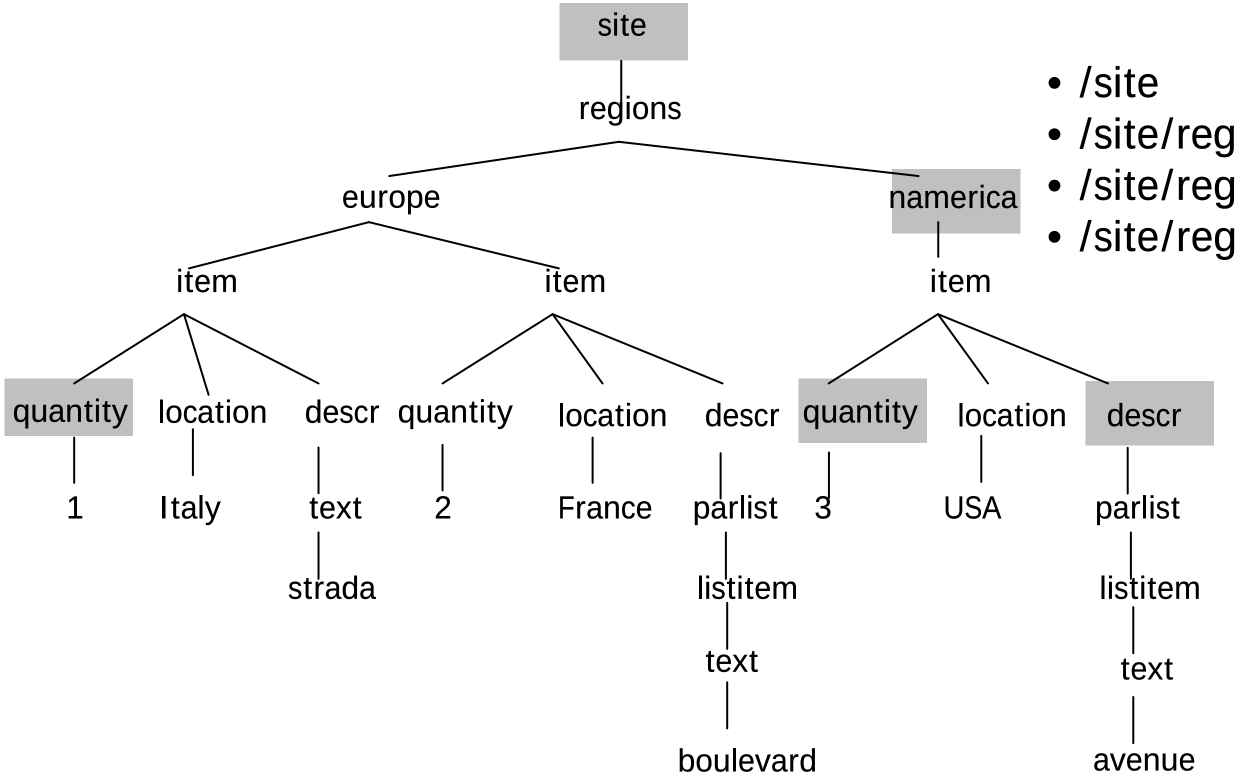
Document D



Fragments of D

Example on XMark (1)

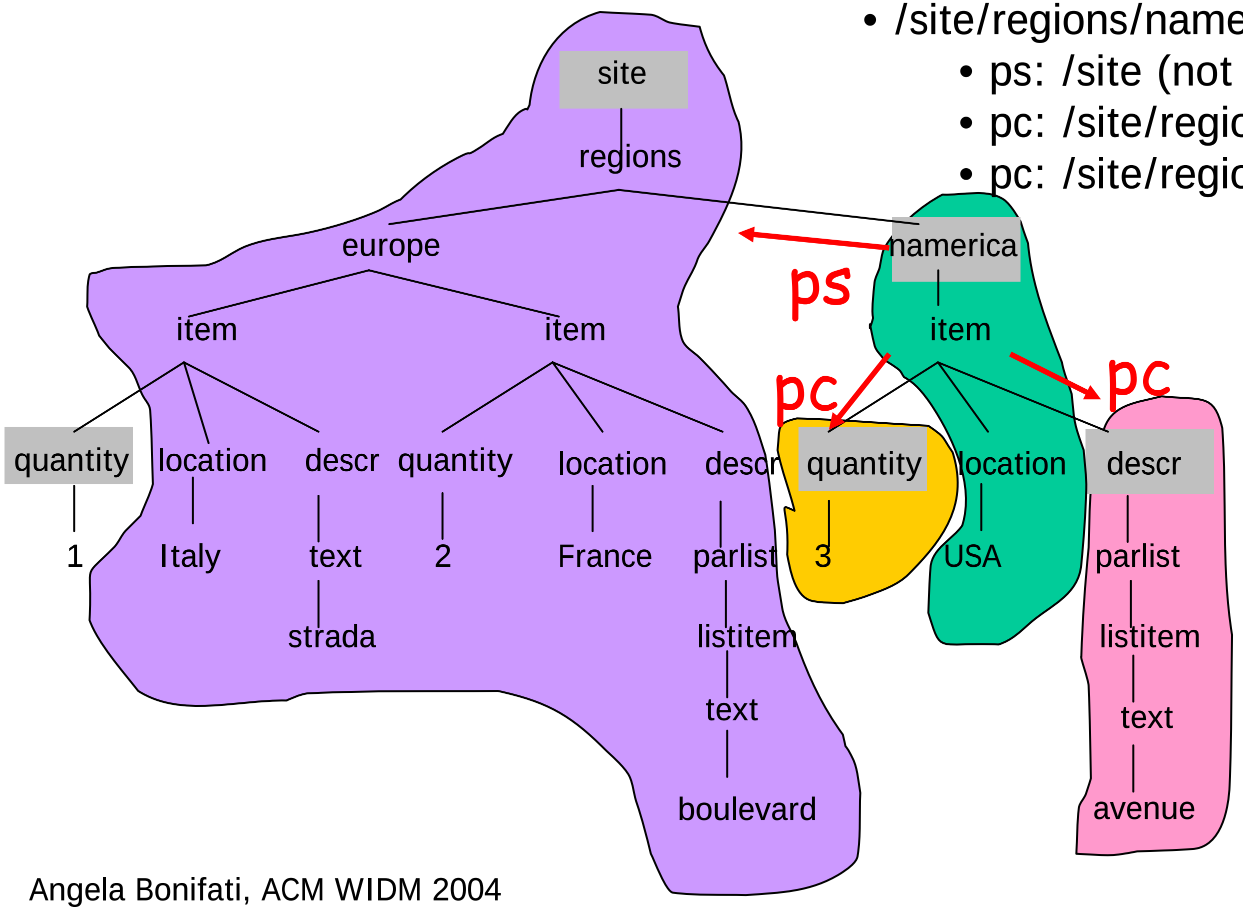
- Given a fragment,
  - The *identifier* of that fragment is the *path* from the document root to the fragment root



- /site
- /site/regions/namerica
- /site/regions/namerica/item[1]/quantity
- /site/regions/europe/item[1]/quantity

Example on XMark (2)

- Given a fragment:
  - – Each fragment maintains the path of its *super* fragment and the paths of its *children* fragments



- /site/regions/namerica
  - ps: /site (not ...regions!)
  - pc: /site/regions/namerica/item[1]/quantity
  - pc: /site/regions/namerica/item[1]/descr

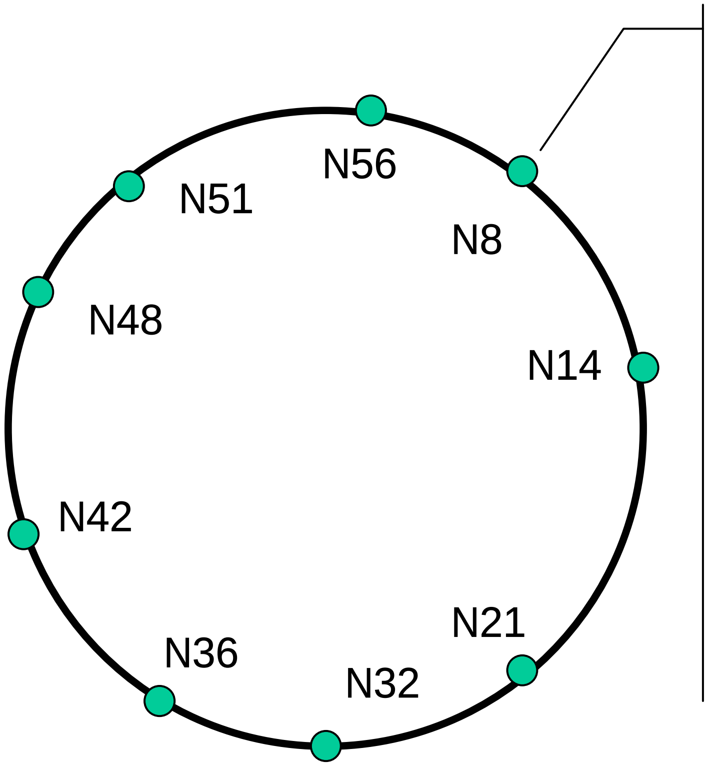
D-hash Paths...

- Paths come as identifiers
  - Each fragment is *accessed* through its own path
  - Super fragments and children fragments are *stored* within the local peer
- Which mechanism to use?
    - A lightweight distributed hash table (DHT)
- Implementation based on Chord (Stoica2001)
  - Maintains list of successors at log distance
  - Guarantees efficient access

XP2P extension of Chord

- XML fragments and Xpaths along the ring:

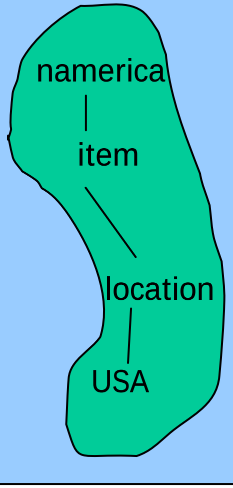
What I need is a mechanism to transform a pc (ps) into a Nx!



- /site/regions/namerica
  - ps: /site
  - pc: /site/regions/namerica/item[1]/quantity
  - pc: /site/regions/namerica/item[1]/descr

Finger Table of N8

N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42



D-hash Paths? Better Fingerprint them

- Hash functions (e.g.SHA-1) are fine but what about...
  - Using a function which cope with updates (frequent in P2P)?
  - Ensure authenticity of data content?
- ...i.e. using fingerprinting [Rabin81]:
  - F(A)=A(t) mod P(t), A binary string, P irreducible polynomial
  - Properties:
    - *F(Concat(/site, /regions))= Concat(/site,F(/regions))*
    - The fingerprinting polynomial is the key:
      - degree equal to 64 means paths of 50 steps, 2^30 fragments
      - policy: you know me if you know my polynomial

Chord extensions

- Path keys instead of id keys
- A limited number of replicas is admitted, ala D-hash:
  - Each successor may want to replicate the fragment
- Fingerprinting instead of hashing
- Storage backend for XML fragments based on (native) Berkeley DB libraries
- *XPath Query Processing over the ring*

Some numbers

- On a network of 10000 XMark fragments (size 3KB-20MB)
  - Path expressions occupancy goes from 100B to 40KB
  - Plus the polynomial (few B)
  - depth is arbitrary (e.g. recursive XMark)
- As opposed to:
  - multilevel Bloom filters (78KB-150KB for 50 elements docs, at most 3 steps paths)[Pitoura2004]
  - combinations of P-Indexes, A-Indexes, T-Indexes (22%-47% of the document) [Bremer2003]

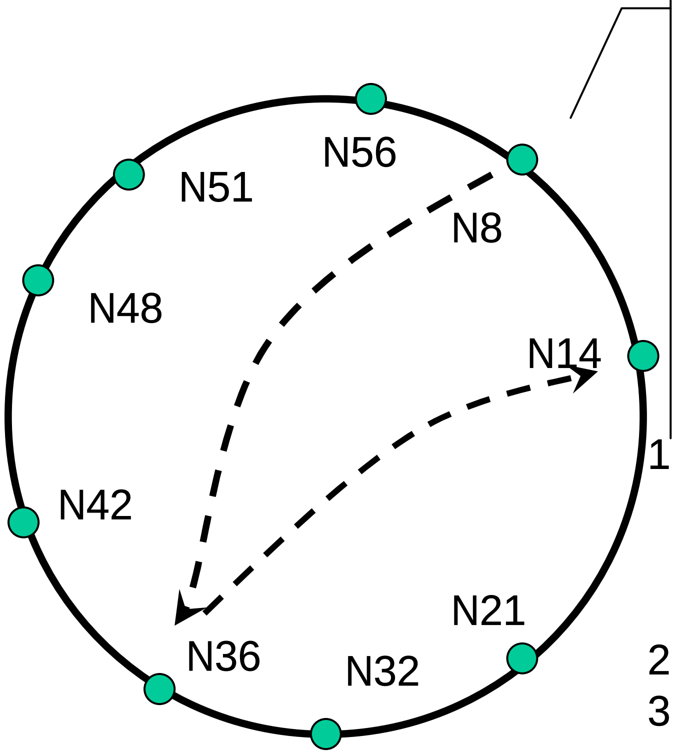
Partial/Full XPath Query Processing

- Queries are partial in XP2P if they imply:
  - Partial answers because of timeout
  - Partial answer because of absence of fragments (peer left)
  - In both cases, the result is approximate

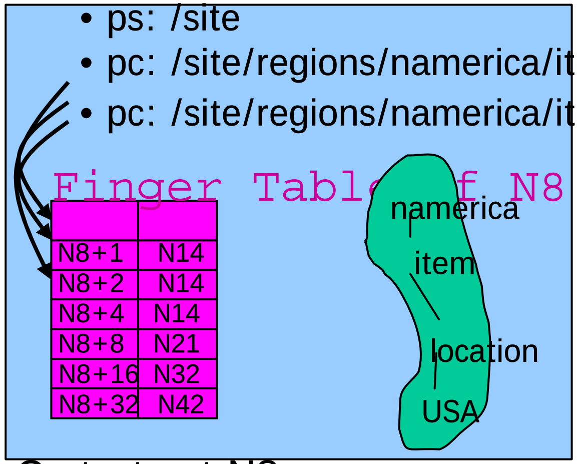
- Full queries may lead to inspect all the network:
  - Differences between queries with/without *descendant* axis
  - Recursion

An algorithm for /,[] queries

- Q: /site/regions/namerica/item[1]/descr/parlist/listitem ?



- /site/regions/namerica
  - ps: /site
  - pc: /site/regions/namerica/item[1]/quantity
  - pc: /site/regions/namerica/item[1]/descr



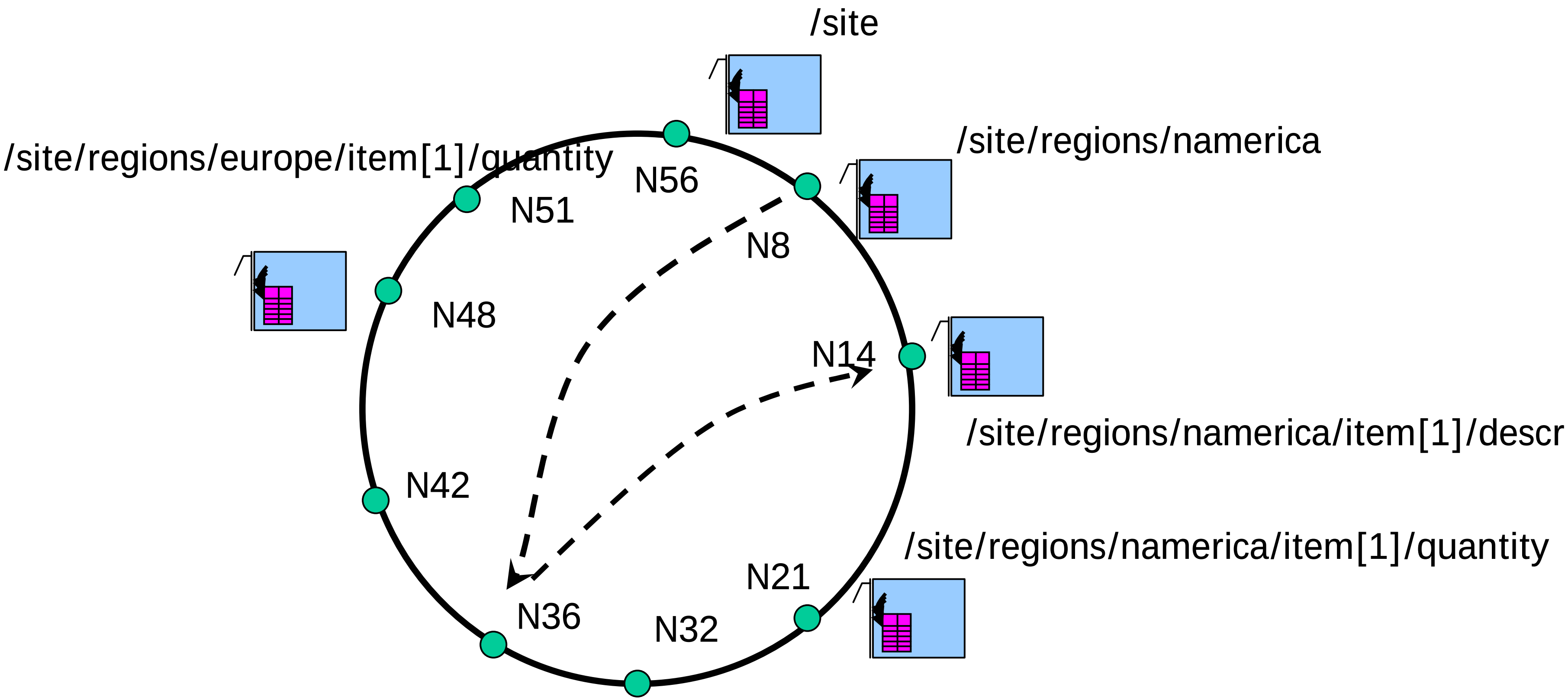
Finger Table of N8

N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

1. Q starts at N8:
   1.1 second pc has same prefix
   1.2 F(Q)= concat(2nd pc, F(/parlist/listitem)) is not present in the ring
2. Locate 2nd pc on FingerTable
3. Jump on resulting peer, if /parlist/listitem is not there, goto next peer and start from 1.1, else return result to N8

An algorithm for // queries(1)

- Q: //item ?


/site
/site/regions/namerica
/site/regions/europe/item[1]/quantity
N56
N51
N8
N48
N14
/site/regions/namerica/item[1]/descr
N42
/site/regions/namerica/item[1]/quantity
N21
N36
N32

Angela Bonifati, ACM WIDM 2004

An algorithm for // queries(2)

- Q: //item ?
- *Step-wise* Algorithm:

```
0. Current peer = N8
1. Q starts at current peer (for each step):
        1.1 local item element on N8 are retrieved
        1.2 search proceeds on pc and ps
2. jump on ps (pc), goto 1 until timeout
```

- Optimization: let p be a *promising* path expression, i.e. a path expression that has the searched element as last step
  - biases search towards particular peers
  - intervenes in 1.2

Experimental Results

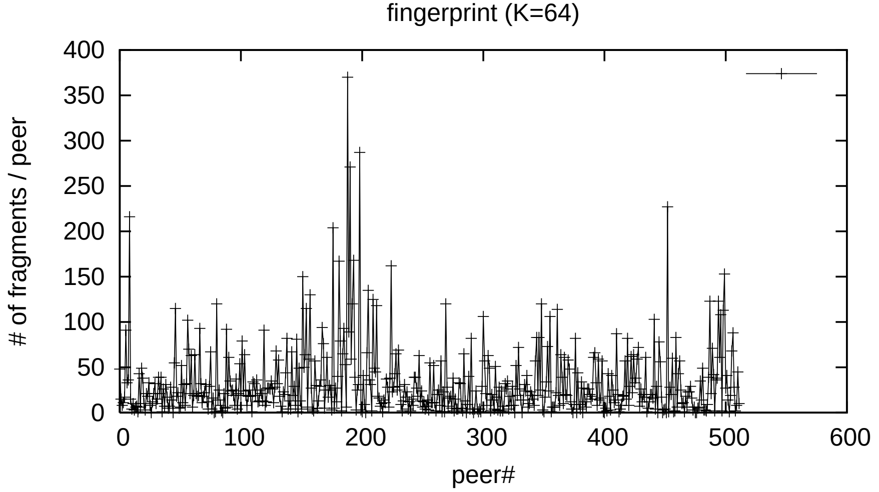
- Fingerprinting:
  - Fingerprinting Load Distribution as opposed to Hash Load Distribution
- Occupancy of related path expressions
- Lookups Performances
  - Partial-match queries
  - Exact-match queries
  - Step-wise algorithm

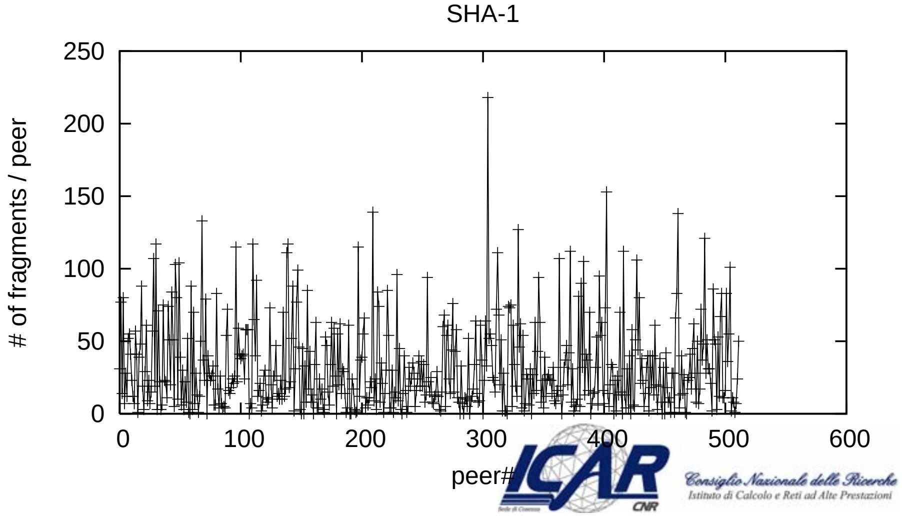
Data sets

- XMark data sets:
  - Small data set:
    - fragments size -> 1KB-1.3MB
  - Medium data set:
    - fragments size -> 3KB-18MB
  - Big data set:
    - fragments size -> 8KB-29MB

Experimental Results(1)

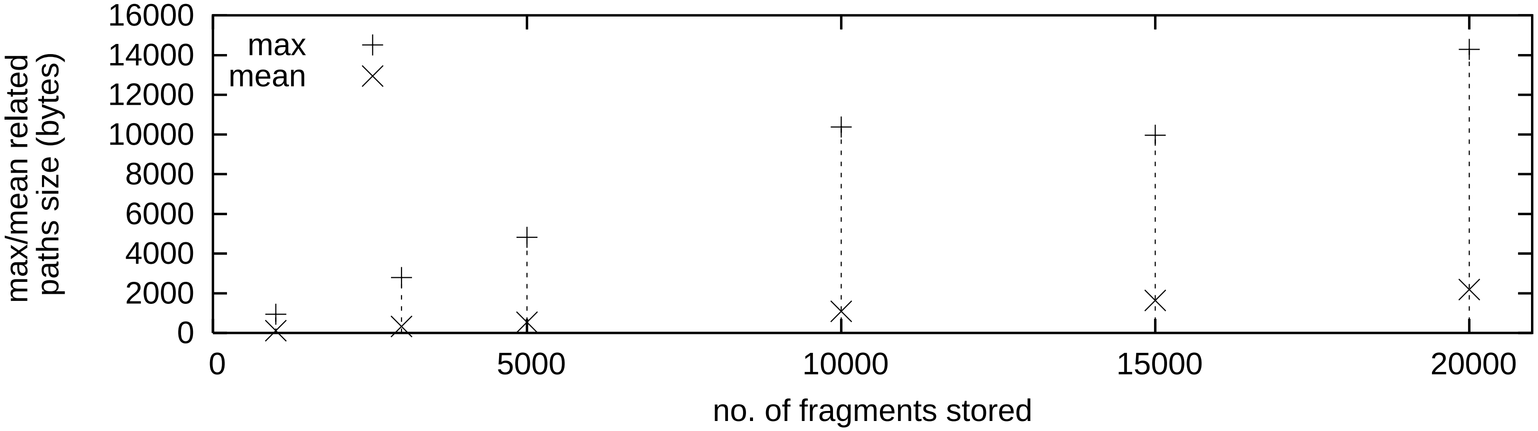
- Simulation with 10000 fragments and 512 peers:





Experimental Results (2)

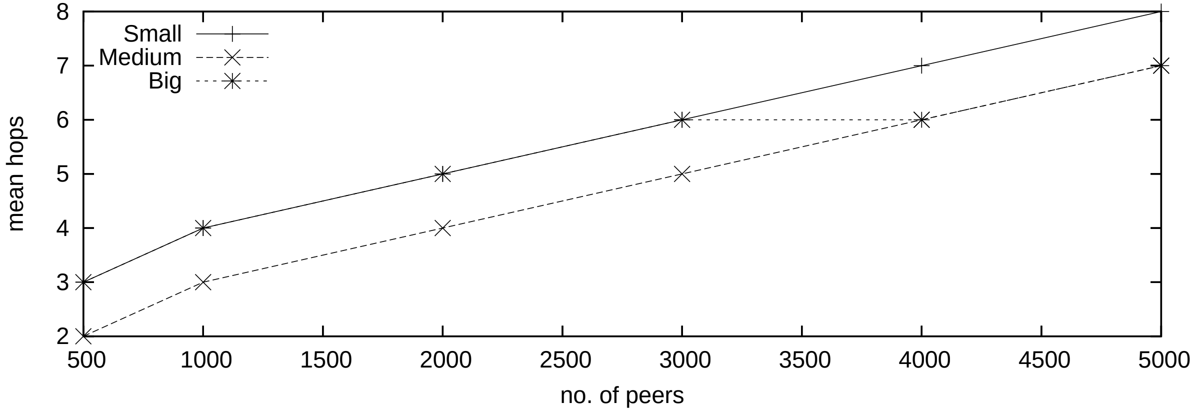
- In a network of 1000 peers, the maximum size is 14KB when 20000 fragments are in the network (*Big* data set).

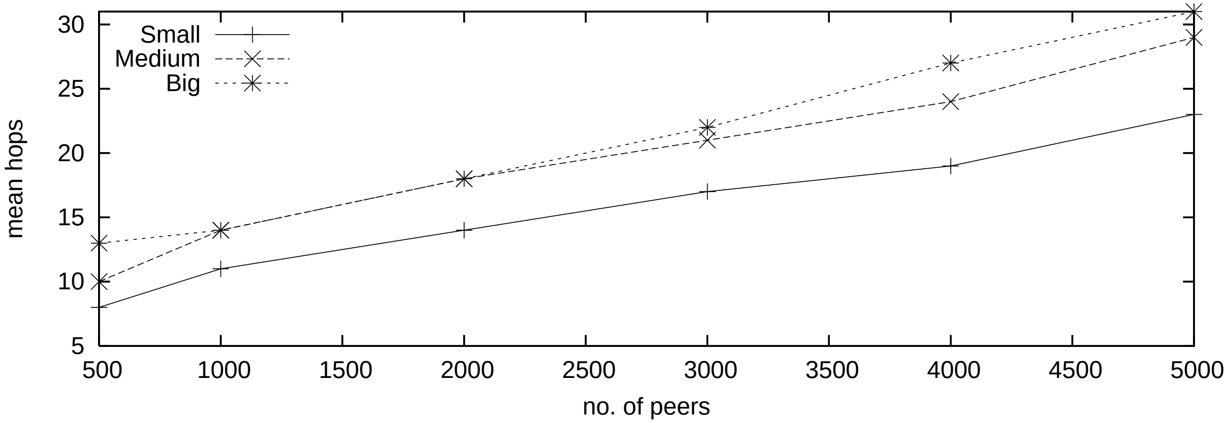


Experimental Results (3)

- Exact-match and partial match, respectively

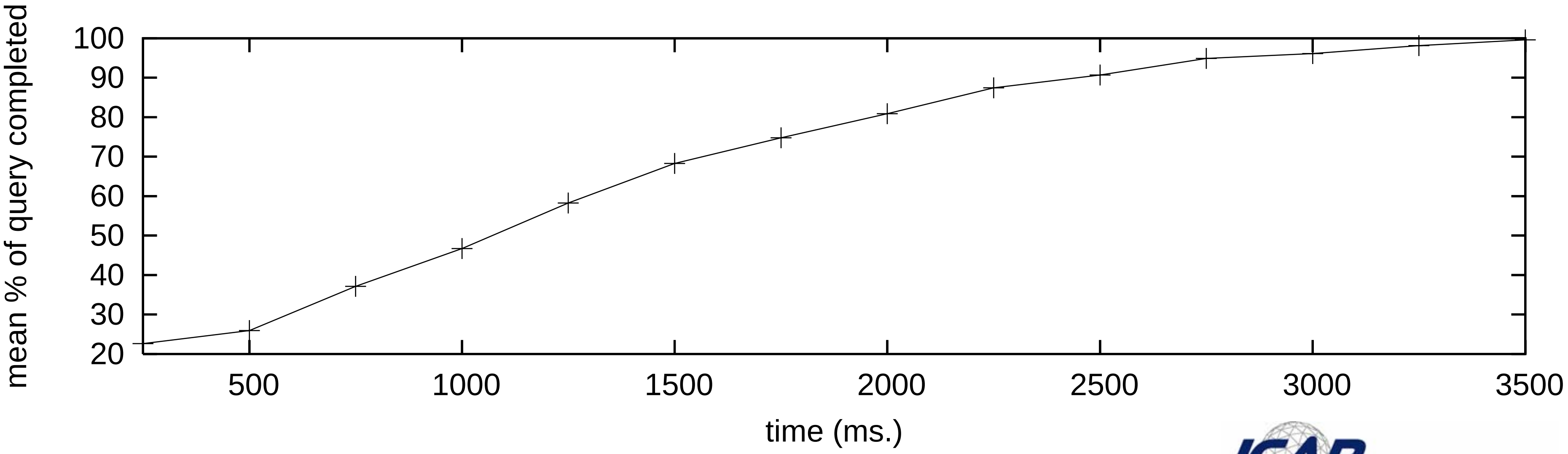
  (queries with /,[]):





Experimental Results (4)

- Step-wise algorithm performances
- 1000 peers network
  with 10000 fragments
  (*Medium* data set)



Conclusions and Future Work

- XP2P is a framework for identifying XML fragments in a P2P network by:
  - Storing a few path expressions and a polynomial on each participating peer
- XP2P achieves reasonable querying times and scalability
- Future work is devoted to:
  - finding optimizations of descendant-axis queries
  - extending the language fragment to deal with (XPath 2.0, XQuery)

Bibliography

- [*Bremer2003*] J.M. Bremer and M. Gertz. *On Distributing XML Repositories.* In WebDB, 2003
- [*ElAbbadi2003*] A.Gupta, D. Agrawal, and A. El Abbadi. *Approximate Range Selection Queries in Peer-to-Peer Systems.* In CIDR, 2003
- [*Galanis2003*] L. Galanis, Y. Wang, S. Jeffery, and D. De Witt. *Locating Data Sources in Large Distributed Systems.* In VLDB, 2003.
- [*Gerhke2004*] A. Craineceanu, P. Linga, J. Gehrke and J. Shanmugasundaram. *Querying Peer-to-Peer Networks using P-trees*. In WebDB 2004.
- [*Loo2004*] B.T. Loo, R. Huebsch, J.M. Hellerstein, I. Stoica and S. Shenker. *Enhancing P2P File-sharing with an Internet-scale Query Processor.* In VLDB 2004.
- [*Pitoura2004*] G. Koloniari and E.Pitoura. *Content-based Routing of Path Queries in Peer-to-Peer Systems*. In EDBT 2004.