Decentralized Information Sharing in Grid Environments

Cristina Schmidt and Manish Parashar
The Applied Software Systems Laboratory
Rutgers, The State University of New Jersey

Siemens, January 2005
Outline

- Introduction to Grid computing
- Squid - Content-based information discovery and sharing
  - Design and implementation
  - Evaluation
- A Collaboratory of TMA data sharing
- Summary
The Grid vision

- Imagine a world
  - in which computational power (resources, services, data, etc.) is as readily available as electrical power
  - in which computational services make this power available to users with differing levels of expertise in diverse areas
  - in which these services can interact to perform specified tasks efficiently and securely with minimum of human intervention
    - on-demand, ubiquitous access to computing, data, and services
    - new capabilities constructed dynamically and transparently from distributed services
  - a large part of this vision was originally proposed by Fenando Corbato (The Multics Project, 1965, www.multicians.org)
Grids – An evolving vision ...

- Seamless aggregation
  - aggregation of capacity
  - aggregation of capability

- Seamless compositions and interactions

- Autonomic behaviors
Enabling Grid Computing - Exponentials

- Network vs. computer performance
  - Computer speed doubles every 18 months
  - Storage density doubles every 12 months
  - Network speed doubles every 9 months
  - Difference = order of magnitude per 5 years

- 1986 to 2000
  - Computers: x 500
  - Networks: x 340,000

- 2001 to 2010
  - Computers: x 60
  - Networks: x 4000

"When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances"

(George Gilder)
The Grid – Seamless aggregation

- Coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.
  - A VO is a collection of users sharing similar needs and requirements in their access to processing, data and distributed resources and pursuing similar goals.
- Key concept:
  - Ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose.
Grid Evolution – From seamless aggregation to seamless composition/interaction

- Electrical Chemical Lab Service
- Computation service
- Image processing service
- Database service

Modalities:
- e.g., MRI, CT, CR,
- Ultrasound, etc.
P2P Data Discovery and Sharing: Motivation

- The need for information discovery in large, decentralized, distributed resource sharing environments, in the absence of global knowledge of naming conventions

- Examples:
  - P2P Document Sharing Systems
  - Grid Resource Discovery
  - Web Service Discovery
Squid: Overview

- Squid is a Peer-to-Peer (P2P) indexing and information discovery system
- Supports complex queries containing partial keywords, wildcards and range queries
- Guarantees that all existing data elements matching a query will be found with bounded cost in terms of number of messages and nodes involved
Related Work

- **Unstructured (Gnutella-like)**
  - Unstructured overlay network, use flooding

- **Hybrid (Napster)**
  - Unstructured overlay network, use centralized directories for search

- **Data-lookup (CAN, Chord, Pastry, etc)**
  - Structured overlay, Internet-scale DHT

- **Structured keyword search**
  - Structured overlay, extend data-lookup protocols
  - Examples:
    - *Distributed Inverted Indices*
    - *Space Filling Curve*
System components:

- Locality preserving mapping that maps documents to indices – using Space Filling Curves (SFC)
- Overlay network of nodes (Chord)
- Load balancing mechanisms
- A query engine that supports complex queries
Document space

- Documents have assigned keywords

2-dimensional keyword space for a P2P sharing system

3-dimensional keyword space for storing computational resources, using the attributes: storage space, base bandwidth and cost
Basic operations:

- Store documents in the system
- Attach keywords to document
- Index the document using Hilbert Space Filling Curve
- Store the document at the appropriate node in Chord

Design - Overview

Document (kw1, kw2, ..., kWd)

Peers (P1, P2, ..., Pk, ...)

Point in a D-dimensional space

Point in a 1-dimensional index space

SFC
Query the system
- Detect the index regions that are defined by the query
- Query the appropriate nodes in Chord overlay
- Get the data
Hilbert Space-Filling Curve (SFC)

- \( f : \mathbb{N}^d \rightarrow \mathbb{N} \), recursive generation

Properties:
- Digital causality
- Locality preserving
- Clustering
Using SFC to generate the index space

- the d-dimensional keyword space is mapped to a 1-dimensional index space using SFC
The overlay network

- Use Chord as overlay network

Cost to look-up data: $O(\log_2 N)$

Overlay network with 5 nodes and an identifier space from 0 to 64

Each node stores the keys that map to the segment of the curve between itself and the predecessor node.
The Query Engine

- Query: combination of keywords, partial keywords, wildcards, ranges
- Example:
  - (computer, network)
  - (computer, net*)
  - (comp*, *)
  - (256-512MB, *, 10Mbps-*) (memory, cost, base bandwidth)
Query Processing

- **Step 1**: Translate the query to relevant clusters on the SFC-based index space

- **Step 2**: Query the appropriate nodes in the overlay

Query, e.g. (computer, *)

Query the nodes 13 and 32
Query optimization

- Not all clusters that are generated for a query exist in the network => optimize!
- SFC generation recursive => clusters generation is recursive => the process of cluster generation can be viewed as a tree
- Optimization: embed the tree into the overlay, and prune nodes during the construction phase
Query optimization – illustration

Solve query: (011, *)
Embed the leftmost tree path (solid arrows) and the rightmost path (dashed arrows) onto the overlay network topology.
Query: *(Staining intensity = 3, Staining area = *) => Binary query (011, *)
Experimental evaluation

- 1000 to 5400 nodes
- Up to $10^6$ keys (unique keyword combinations)
- Metrics:
  - Number of routing nodes
  - Number of processing nodes
  - Number of data nodes
  - Number of messages
- Query types:
  - Q1: (computer, *), (comp*, *, *)
  - Q2: (comp*, net*), (computer, network, *)
  - Q3: range queries
2D space – Q1 and Q2 queries

- System size increases from 1000 to 5400 nodes, keys from $2 \times 10^5$ to $10^6$
3D space – Q1 and Q2 queries
3D space – range queries
Load balancing

- **Load balancing at node join:**
  - generate more than one ID for the new node, send join requests in the network and join with the ID that places the node in the most crowded part of the network

- **Load balancing at runtime:**
  - run a local load balancing algorithm between neighbors (from time to time), and redistribute the load
  - use virtual nodes that can migrate to less loaded physical nodes
Load balancing

The distribution of the keys in the index space. The index space was partitioned into 5000 intervals. The Y-axis represents the number of keys per interval.

The distribution of the keys when using only the load balancing at node join technique.

The distribution of the keys when using both the load balancing at node join technique, and the local load balancing.
The P2P TMA Collaboratory

- Imaging, analysis, and sharing of tissue microarrays (TMAs), correlated clinical data and experimental results
- Consortium of distributed clinical and research sites
- Goal: facilitate cooperative oncology research
Definitions

- **TMA – Tissue MicroArray**
Challenges

- Analysis and evaluation of TMA samples
- Data management - large volume of TMA data
- Tens to hundreds of samples on a slide
- A digitized TMA specimen of app. 400 discs = 18GB
- Sharing the data
- Ownership issues
- A standard data representation
Motivation – why P2P?

Why a structured P2P with flexible queries and guarantees?

- Scale – large number of participants, and large quantities of data
- Data owned by research centers and hospitals – only the index is distributed
- Need guarantees – “rare” data that exists in the system has to be found by a matching query
- Flexible queries – wildcards and ranges
P2P TMA Collaboratory - Overview

P2P System (Squid)

Indexing Peer
Publish and Query
SFC Index
P2P Overlay

User Interface

Data gathering, processing and archiving
P2P TMA Collaboratory - Overview

Clinical history

Construction and preparation of TMA slides

TMA Slide

Data gathering

Imaging and diagnostic (TMA-AID)

TMA Data

Data processing

Array Archiving (AA)

Database

Metadata extraction

Distributed Telemicroscopy (DT)

Database Portal

Data access

Publish

Remote access to TMA-AID module

Metadata extraction module

Data discovery

Squid

Publish and Query

SFC index

P2P overlay

User Interface

Database Array

Archiving (AA)

Remote access to TMA-AID module

Metadata extraction

Siemens, January 2005
Publishing and finding data

- **Publish**
  - Database Table
  - Metadata (XML)
  - Squid Index
  - Squid

- **Search for data**
  - 1. User query
  - 2. Query Squid
  - 3. Query results
  - 4. Data exchange
  - Database Portal

 Siemens, January 2005
Basic Squid operations:
- Publish TMA metadata, using keywords
- Discover TMA metadata, using keywords, wildcards, ranges

Uses the Space Filling Curves for indexing, and Chord as an overlay network
Hilbert Space-Filling Curve (SFC), Chord

**Properties:**
- Digital causality
- Locality preserving
- Clustering

**SFC - f: \( N^d \rightarrow N \), recursive generation**

**Chord P2P overlay**

**Properties:**
- Each node has a unique identifier
- Each node maintains a "finger" table
- Data lookup in \( O(\log n) \)
Squid – publishing metadata example

- Publish – data representation

```xml
<tma_entry>
  <CancerType>breast cancer</CancerType>
  <Age>61</Age>
  <DrugRegimen>Arimidex</DrugRegimen>
</tma_entry>
```

- Keywords: breast cancer, 61, Arimidex
Squid – publishing TMA metadata

1. Extract keywords

(staining intensity = 2, staining area = 1)

2. The TMA metadata file is described by keywords

3. Map the point (2, 1) to index 7 using the Hilbert Space Filling Curve (SFC).

4. Store the metadata file at node 13 (the successor of the index 7)
Squid – querying metadata example

- Information discovery – query representation

```xml
<tma_query>
  <CancerType>breast cancer</CancerType>
  <Age>30-40</Age>
  <DrugRegimen>*</DrugRegimen>
</tma_query>
```

- The query: “breast cancer”, age between 30 and 40, any drug regimen
Squid – retrieving TMA metadata

Query 1:
(staining intensity = (4..7), staining area = (0...3))

Query 2:
(staining intensity = *, staining area = 4)

1. Formulate query

2. Generate the clusters associated with the query

3. Query the nodes that store the clusters

4. Send the results to the user

TMA Metadata files for Query 1

TMA Metadata files for Query 2
Experimental evaluation

- Simulation
  - 1000 to 5400 nodes and up to $10^6$ metadata files, characterized by unique keyword combinations
  - Query types:
    - Q1: (her2, *, *)
    - Q2: (breast cancer, amiridex, *)
    - Q3: range queries
      - Q3_1: (keyword, range, *)
      - Q3_2: (range, range, range)
  - Measured the number of nodes that process the query, and the number of nodes that found data matching the query
Simulation, 3D, for Q1 and Q2

Simulation, 3D, range queries
Evaluation

- Implementation on JXTA (a general P2P framework)
  - 64 node Linux cluster, 1.6 GHz Pentium IV, each node running a peer
  - Measured the overhead to process a query at a node

Query processing overhead at a node
Load balancing at node join:
- generate more than one ID for the new node, send join requests in the network and join with the ID that places the node in the most crowded part of the network

Load balancing at runtime:
- run a local load balancing algorithm between neighbors (from time to time), and redistribute the load
- use virtual nodes that can migrate to less loaded physical nodes
**Load balancing**

The distribution of the keys in the index space. The index space was partitioned into 5000 intervals. The Y-axis represents the number of keys per interval.

The distribution of the keys when using only the load balancing at node join technique.

The distribution of the keys when using both the load balancing at node join technique, and the local load balancing.

Siemens, January 2005
Summary

Squid infrastructure for information discovery in Grid environments guarantees and bounded costs flexible querying structured overlay Design of a prototype P2P Collaboratory for imaging, analyzing and sharing of TMA data

in Grid environments guarantees and bounded costs flexible querying structured overlay

Summary
Questions