PeerSpaces: Data-Driven Coordination on P2P Networks (2003)

From Nadia Busi, Cristian Manfredini, Alberto Montresor and Gianluigi Zavattaro of University of Bologna

Presented by Samuele Milani
Summary

- New coordination model (The Peer Spaces Model)
- Exploit the Linda Model for Peer-to-Peer Networks.
- Prototypical implementation (JXTA).
Overview

1) Peer-to-Peer Networks
2) The Linda Model
3) Linda in a Mobile Environment (LIME)
4) Peer Spaces
5) An Implementation with JXTA
1. Peer-to-Peer Networks
Peer-to-Peer Basics

- Various type of Networks:
  - Client-Server: Web servers
  - Hybrid P2P: Napster, eDonkey
  - Pure P2P: Gnutella, Freenet
The eDonkey Network

Concurrency Seminar: PeerSpaces Data-Driven Coordination on P2P Networks
Problems?

- High communication overhead
- Slow search
- Distribution of load & resources
- Elephant & Mice problem
- ...

Concurrency Seminar: PeerSpaces Data-Driven Coordination on P2P Networks
2. The Linda Model
The Linda Model

- Developed at Yale University in 1985
- **Shared memory** accessible by all processes
- Data elements consists of **tuples**
- Tuple is like a record ex. (`worker', 0, i)
- Four basic primitives
- Implemented through libraries (Linda for C, Fortran, C++)
out('test', 4, 64)
   places a data tuple in tuple space

eval('test', i, f(i))
   creates a live tuple, which start a process that evaluate each argument

rd('test', ?i, ?j) blocking
   read the values in a tuple in a tuple space

in('test', ?i, ?j) blocking
   same as rd except that it also removes the tuple from the tuple space
Example

**Tuple Space**

- \( \langle \text{Car}, 180, \text{Volvo} \rangle \)
- \( \langle \text{Test}, 1 \rangle \)
- \( \langle \text{Test2}, 18 \rangle \)

- `out('Job1', 100, '$')`

**Concurrency Seminar: PeerSpaces Data-Driven Coordination on P2P Networks**
Example

Tuple Space

- ("Car", 180, "Volvo")
- ("Test", 1)
- ("Test2", 18)
- ("Job1", 100, ")"

Concurrent Seminar: PeerSpaces Data-Driven Coordination on P2P Networks
Example

```
Example

tuple space

('Car', 180, 'Volvo')

('Test', 1)

('Test2', 18)

('Job1', 100, '$_$')

Concurrency Seminar: PeerSpaces Data-Driven Coordination on P2P Networks
```
3. Lime

(Linda in Mobile Environment)
Lime

- Problems with Linda:
  - where I store the tuples (Server)
  - rather statical model
  - not suitable for dynamical changing networks
- Lime designed for mobile applications over wired and ad hoc networks
- Supports physical and logical mobility of the hosts
- Introduces transiently shared data space
- Decoupled and opportunistic style of computation
Transiently shared Dataspaces

Process Level
Data Space

Host Level
Data Space

Transiently shared
Data Space
4. Peer Spaces
Motivation

- Adapt the Linda/Lime Model to P2P
- Generative communication
- Must support extreme dynamism in structure, content and load
- Large degree of self-configuration and self management is required
- New coordination model PeerSpaces
- Formal rules, abstraction of data and network structure
Definitions

- Data (d)
  - Local data $d$
  - Generic data $d_g$ (Chat)
  - Replicable data $d_r$ (FileSharing)
- Peer p [P, DS]
- Ps set of peers
- MD Misplaced data
- Network topology △ △
A Peer Space Example

Peer p:
p(\text{write}(\text{job}, \text{Gen}) \cdot \text{read}(\text{response}, 8) \cdot \text{write}(\text{cash}, \text{Gen}) \cdot P, \{\})

Peer q:
q(\text{read}(\text{job}, 8) \cdot \text{write}(\text{response}, \text{Rep}) \cdot \text{read}(\text{cash}, 8) \cdot \\
\text{write}(\text{pay}, \text{Gen}) \cdot Q, \{\text{dog}\})

Peer r:
r(\text{read}(\text{pay}, 8) \cdot \text{write}(\text{receipt}, \text{Gen}) \cdot R, \{\text{credit}\})

Network:
- fully connected

MD = \{\}
A Peer Space Example

Peer p:
\[ p(\text{write}(\text{job}, \text{Gen}). \text{read}(\text{response}, 8). \text{write}(\text{cash}, \text{Gen}). P, \{\}) ) \]

Peer q:
\[ q(\text{read}(\text{job}, 8). \text{write}(\text{response}, \text{Rep}). \text{read}(\text{cash}, 8). \text{write}(\text{pay}, \text{Gen}). Q, \{\text{dog}\}) ) \]

Peer r:
\[ r(\text{read}(\text{pay}, 8). \text{write}(\text{receipt}, \text{Gen}). R, \{\text{credit}\}) ) \]

Network:
- fully connected
MD = \{ \}
A Peer Space Example

Peer p:
\[
p(\text{read}(\text{response}, 8). \text{write}(\text{cash}, \text{Gen}). P, \{\text{job}_g\})
\]

Peer q:
\[
q(\text{read}(\text{job}, 8). \text{write}(\text{response}, \text{Rep}). \text{read}(\text{cash}, 8).
\hspace{1cm} \text{write}(\text{pay}, \text{Gen}). Q, \{\text{dog}\})
\]

Peer r:
\[
r(\text{read}(\text{pay}, 8). \text{write}(\text{receipt}, \text{Gen}). R, \{\text{credit}\})
\]

Network:
- fully connected

MD = { }
A Peer Space Example

Peer p:
\[ p(\text{read}(\text{response},8).\text{write}(\text{cash},\text{Gen}).P, \{\}) ) \]

Peer q:
\[ q(\text{read}(\text{job},8).\text{write}(\text{response},\text{Rep}).\text{read}(\text{cash},8). \]
\[ \quad \text{write}(\text{pay},\text{Gen}).Q, \{\text{dog}\}) \]

Peer r:
\[ r(\text{read}(\text{pay},8).\text{write}(\text{receipt},\text{Gen}).R, \{\text{credit}\}) \]

Network:
- fully connected

\[ \text{MD} = \{\text{job}_g\} \]
A Peer Space Example

Peer p:
p(read(response, 8).write(cash, Gen).P, { })

Peer q:
q(write(response, Rep).read(cash, 8).
  write(pay, Gen).Q, {dog, jobg})

Peer r:
r(read(pay, 8).write(receipt, Gen).R, {credit})

Network:
- fully connected
MD = { }
A Peer Space Example

Peer p:
\[ p(\text{read}(\text{response}, 8) . \text{write}(\text{cash}, \text{Gen}) . P, \{ \}) \]

Peer q:
\[ q(\text{read}(\text{cash}, 8) . \text{write}(\text{pay}, \text{Gen}) . Q, \{\text{dog, job, response}\}) \]

Peer r:
\[ r(\text{read}(\text{pay}, 8) . \text{write}(\text{receipt}, \text{Gen}) . R, \{\text{credit}\}) \]

Network:
- fully connected
MD = \{ \}
A Peer Space Example

Peer p:
p(\texttt{write(cash,Gen).P, \{response\}})

Peer q:
q(\texttt{read(cash,8)}).
\hspace{1cm} \texttt{write(pay,Gen).Q, \{dog, job, response\}}

Peer r:
r(\texttt{read(pay,8).write(receipt,Gen).R, \{credit\}})

Network:
- fully connected

MD = \{ \}
A Peer Space Example

Peer \( p \) :
\[ p(P, \{\text{response, cash}_g\}) \]

Peer \( q \) :
\[ q(\text{read}(\text{cash}, 8) . \text{write}(\text{pay}, \text{Gen}).Q, \{\text{dog, job, response}\}) \]

Peer \( r \) :
\[ r(\text{read}(\text{pay}, 8).\text{write}(\text{receipt}, \text{Gen}).R, \{\text{credit}\}) \]

Network :
- fully connected

MD = \{\}
A Peer Space Example

Peer p:
p(P, {response})

Peer q:
q(
    \texttt{write(pay, Gen).Q, \{dog, job, response, cash\}}
)

Peer r:
r(\texttt{read(pay, 8).write(receipt, Gen).R, \{credit\}})

Network:
- fully connected

MD = \{\}

\textit{Concurrency Seminar: PeerSpaces Data-Driven Coordination on P2P Networks}
A Peer Space Example

Peer p :
p(P, {response})

Peer q :
q( Q, {dog, job, response, cash, pay} )

Peer r :
r(read(pay, 8).write(receipt, Gen).R, {credit})

Network :
- fully connected

MD = { }
A Peer Space Example

Peer p :
p(P, \{\text{response}\})

Peer q :
q(
   Q, \{\text{dog, job, response, cash}\}
)

Peer r :
r(\text{read(pay, 8).write(receipt, Gen).R, \{credit\}})

Network :
- A Node has disconnected now Route(q,r) = false
MD = \{\text{pay}_g\}
Data Production

(1) \( (p[\text{write}(d, \text{Here}).P, DS] \oplus P_s, \bowtie, MD) \rightarrow (p[P, DS \oplus d] \oplus P_s, \bowtie, MD) \)

(2) \( (p[\text{write}(d, \text{Gen}).P, DS] \oplus P_s, \bowtie, MD) \rightarrow (p[P, DS \oplus d_g] \oplus P_s, \bowtie, MD) \)

(3) \( (p[\text{write}(d, \text{Rep}).P, DS] \oplus P_s, \bowtie, MD) \rightarrow (p[P, DS \oplus d_r] \oplus P_s, \bowtie, MD) \)

(4) \( (p[\text{write}(d, p').P, DS] \oplus P_s, \bowtie, MD) \rightarrow (p[P, DS] \oplus P_s, \bowtie, MD \oplus \langle d \rangle_{p'}^p) \)

(5) \( (p'[P, DS] \oplus P_s, \bowtie, MD \oplus \langle d \rangle_{p'}^p) \rightarrow (p'[P, DS \oplus d] \oplus P_s, \bowtie, MD) \quad \text{if Route}(p, p') \)
(6) \( (p[\text{read}(d, h).P, DS \oplus d'] \oplus Ps, \varpi, MD) \rightarrow (p[P, DS \oplus d'] \oplus Ps, \varpi, MD) \) if \( d' \in \{d, d_g, d_r\} \)

(7) \( (p[\text{read}(d, h).P, DS] \oplus p'[P', DS' \oplus d'] \oplus Ps, \varpi, MD) \rightarrow (p[P, DS] \oplus p'[P', DS' \oplus d'] \oplus Ps, \varpi, MD) \) if \( d' \in \{d, d_g, d_r\} \land p' \in \text{Hor}(p, h) \)

(8) \( (p[\text{take}(d, h).P, DS \oplus d'] \oplus Ps, \varpi, MD) \rightarrow (p[P, DS] \oplus Ps, \varpi, MD) \) if \( d' \in \{d, d_g\} \)

(9) \( (p[\text{take}(d, h).P, DS] \oplus p'[P', DS' \oplus d'] \oplus Ps, \varpi, MD) \rightarrow (p[P, DS] \oplus p'[P', DS'] \oplus Ps, \varpi, MD) \) if \( d' \in \{d, d_g\} \land p' \in \text{Hor}(p, h) \)
Constant and Context Rules

\[ (10) \quad \Box \leftrightarrow \Box' \]
\[ (P_s, \Box, MD) \rightarrow (P_s, \Box', MD) \]

\[ (11) \quad (p[P, DS] \oplus P_s, \Box, MD) \rightarrow P_s' \]
\[ (p[K, DS] \oplus P_s, \Box, MD) \rightarrow P_s' \quad \text{if} \quad K = P \]

\[ (12) \quad (p[P, DS \oplus d_g] \oplus p'[P', DS'] \oplus P_s, \Box, MD) \rightarrow \]
\[ (p[P, DS] \oplus p'[P', DS' \oplus d_g] \oplus P_s, \Box, MD) \quad \text{if} \quad \text{LoadBal}(p, p', d_g) \]

\[ (13) \quad (p[P, DS \oplus d_r] \oplus p'[P', DS'] \oplus P_s, \Box, MD) \rightarrow \]
\[ (p[P, DS \oplus d_r] \oplus p'[P', DS' \oplus d_r] \oplus P_s, \Box, MD) \quad \text{if} \quad \text{LoadBal}(p, p', d_r) \]
Open issues?

- Implementation of:
  - Route \((p, p')\)
  - Hor \((p, h)\)
  - LoadBal \((p, p', d)\)
5. A JXTA Prototypical Implementation
What’s JXTA?

- JXTA is an open-source project by Sun to provide a set of basic facilities for P2P applications
- Based on a set of XML protocols
- JXTA middleware has 3 layers
  - core: low-level functions (routing, communication, ...)
  - services: indexing, searching, file sharing
  - applications: high-level application (chat, auction, ...)
The JPS Network

Concurrency Seminar: PeerSpaces Data-Driven Coordination on P2P Networks
Conclusion

- We can not yet draw conclusions until concrete implementations comes out !!!!
- But the model should theoretically work

To be continued...