Communication Facilities for Database Transaction Processing

by
Prof. Bharat Bhargava
Dept. of Computer Science
Purdue University
West Lafayette, IN 47907
USA

317-494-6013
317-494-0739 (Fax)
bb@cs.purdue.edu
Table of Content

- Statement of problem
- Assumption
- Transaction processing benchmark
- RAID distributed database system
- Related projects
- LAN communication for DTP
- Evaluation of RAID communication
- WAN communication for DTP
Legend

**DTP** Distributed Transaction Processing

**LAN** Local Area Network

**WAN** Wide Area Network

**IPC** Inter-Process Communication

**RPC** Remote Procedure Call
Statement of the Problem

How to control and improve the performance of distributed transaction processing (DTP) system by improving the communication and by exploiting the knowledge about network characteristics, architecture, and dynamics?
Assumptions

- DTP is communication intensive.

- The conventional communication scheme cannot meet the performance requirement.

- Wide area networks are characterized by large number of sites, variable communication delay, and high message loss rate etc.

- Distributed transaction processing experiences a serious performance degradation during failures or changing of communication conditions.

- It's difficult to conduct experiments and testing in WAN environments.
## LAN vs WAN

<table>
<thead>
<tr>
<th>Issues</th>
<th>LAN</th>
<th>WAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>small (10-100 sites)</td>
<td>large (over 100 sites)</td>
</tr>
<tr>
<td>Geographic span</td>
<td>within a mile</td>
<td>over 100 miles</td>
</tr>
<tr>
<td>Topology</td>
<td>bus, ring</td>
<td>hierarchical interconnected irregular mesh</td>
</tr>
<tr>
<td>Routing</td>
<td>simple</td>
<td>multiple hops, dynamic</td>
</tr>
<tr>
<td>Speed</td>
<td>very high (10-1Gbps)</td>
<td>low (1Kbps-10Mbps)</td>
</tr>
<tr>
<td>Error rate</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Variation*</td>
<td>small</td>
<td>large</td>
</tr>
<tr>
<td>Ownership</td>
<td>private</td>
<td>public</td>
</tr>
<tr>
<td>Access</td>
<td>controlled</td>
<td>no control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>shared</td>
</tr>
</tbody>
</table>

* variation of parameters such as delay, throughput, and losses.
Motivation

To improve the performance of DTP by efficient communication facility.

To provide structure as well as efficiency.

To control and improve the performance of DTP by surveillance that exploits the knowledge about network characteristics, architecture, and dynamics.
Transaction Processing Benchmark for Measurements

- Relation size: 100 tuples.
- Hot-spot size: 20% of the tuples.
- Hot-spot access: 80% of the actions access hot-spot tuples.
- Type of experiment: close (maintaining the concurrency level constant).
- Number of transactions: 250.
- Timeout: one second per action.
- Updates: 10% of the actions are updates.
- Restart policy: rolling restart backoff.
- Concurrency controller: two phase locking protocol.
- Atomicity controller: two phase commit protocol.
- Replication controller: ROWA.
- Concurrency level: one transaction at a time.
RAID Experimental Infrastructure

- RAID laboratory with 5 Sun3/50's and 4 Sparcstation1's, all with local disks, microsecond resolution timers connected by a 10Mb/s Ethernet

- Software to specify, create, and maintain replicated databases

- Extended DebitCredit benchmark (Anon 85):
  - Transaction length, and number of transactions to generate
  - Transaction arrival rate or maximum degree of multiprogramming
  - Ratio of small to large transactions
  - Fraction of actions that are updates
  - Hot spot size and access percent

- Action Driver Simulator:
  - Interprets commands written in our benchmark language
- Generates open- and closed- system transactions

- Automated overnight experimental procedure:
  - Reboot machines at 3:00 am
  - Setup environment
    * Unmount non-essential file systems
    * Start up the Oracle (the naming server)
    * Initialize database relations
  - Execute experiments found on the Benchmark directory. For every specified experiment:
    * Start a RAID instance
    * Start the AD simulator to run transactions
    * Terminate the RAID instance
    * Upon termination, RAID servers dump statistics data
  - Next day: digest statistics data into condensed tabular format
The RAID System

Robust and Adaptable Distributed Database System, Versions 1 and 2

- functional distributed database system, 30k lines of C code, running on UNIX
- experimental system for investigating reliability and adaptability
- modular: independent servers connected by location-transparent UDP/IP-based communication package
- individual servers can adapt between algorithms

RAID Version 2
RAID Virtual Sites

Physical sites

Host 1 | Host 2 | Host 3
---|---|---
AC | AM | virtual site 2 | CC | RC
AC | CC | AM | UI1 | virtual site 3 | RC | UI2
AC | AM | RC | UI1 | virtual site 4 | CC

virtual site 1: AC UI1 AM AD1 CC RC
virtual site 2: AC AM
virtual site 3: AM UI1 AD1
virtual site 4: RC UI1 AD1
Distinguishing Features of RAID-V2

- Improved flow of control for adaptability and fault-tolerance

- Directory-based partial replication

- Adaptable quorum-based replication control:
  - An adaptable version of the Quorum Consensus (Gifford 79), in which quorum assignments are determined by a quorum relation

- Dynamically adaptable concurrency control

- Use of XDR to support communication in a heterogeneous system
Adaptability in RAID

- Concurrency Control
- Replication Control
- Distributed Commit
- Network Partitioning
- Reconfiguration after Site Failure
- Inter-server Communication
- Server Relocation
- Parallel vs. Merged Servers (P-Raid)
- Object-relation database model (O-Raid)
- Kernel Extensions (PUSH)

IEEE Transactions on Knowledge and Data Engineering, December 1989.
Related Projects

- Networking/Communication
  - internetworking
  - transport protocols
- Distributed File Systems
  - Cedar, Sprite, Locus, etc
  - Andrew/Coda file systems
- Distributed Operating Systems
  - Amoeba, Athena, Mach, V, x-kernel
  - authentication server (Kerberos), group communication
- Database/Transaction Processing
  - Camelot, Argus, Eden, ISIS, Raid
  - Federated databases, multi-databases (R*)
  - Advanced transaction models
Networking/Communication

• IPC in Distributed Systems
  – Accent - IPC in kernel
  – Mach - hand-off scheduling
  – Camelot - a DTP build on top of Mach
  – VMTP - support intra-system model of DTP
  – DUNE - dynamic binding in RPC

• Local IPC
  – Lightweight RPC
  – User-level RPC

• Communication Protocols and LAN
  – VMTP
  – Virtual/layered protocol (x-kernel)
  – atomic broadcast, multicast

• Communication Support for DTP
  – Camelot, RPC
Distributed Operating System

Amoeba

(Vrije, Tanenbaum)

- Employs transaction model.

- Improves performance by using efficient RPC and local protocols.
  (1.10ms on a 3-MIPS machine)

- Uses "Amoeba gateway" to provide transparent communication in WANs without affecting the performance of RPCs in LANs.

- Fast Local Internet Protocol (FLIP).
Distributed File System

Andrew/Coda File System

(CMU, Satyanarayanan)

• Scalability as the design goal

• Improve performance by volume replication and client caching

• Improve performance by segregating files and access by their semantics

• Improve reliability by distinguishing "disconnected" and "connected" modes
LANs Communication for DTP

- Approaches

- Problems

- Solutions

- Our experiences
General Approaches for LANs

- Multi-threaded, light-weight process
  (Camelot)

- Multi-tasking
  (for multiprocessor machines)

- Merged servers
  (sacrifice the structure)

- Efficient RPC
  (Amoeba, Sprite, V, x-kernel)

- Primitives for transaction management
  (Mach)
Our Approaches for LANs

- Lightweight protocol
- Simple naming scheme
- Memory mapping and shared memory
- Physical multicasting
- Explicit control passing
- Adaptable to the communication environment
Experiments with Conventional Communication Schemes

- Socket-based Interprocess Communication

- Local Communication

- Multicast Communication

- Impact on Transaction Processing
Problem with Conventional Communication Schemes

- General purpose communication facilities are expensive

- DTP systems: communication intensive, most are local rather than remote

- Lack of communication support for DTP in some OS

- General purpose multicasting is unusable

- Name resolution is too expensive
Timing diagram for UDP and SE datagram services.
Local IPC Measurements

<table>
<thead>
<tr>
<th>METHOD</th>
<th>MESSAGE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 Bytes Time (ms)</td>
</tr>
<tr>
<td>2 Q Message Passing</td>
<td>2.0</td>
</tr>
<tr>
<td>1 Q Message Passing</td>
<td>2.0</td>
</tr>
<tr>
<td>Named Pipes</td>
<td>2.3</td>
</tr>
<tr>
<td>Shared Memory</td>
<td>5.1</td>
</tr>
<tr>
<td>UDP Communication</td>
<td>4.3</td>
</tr>
</tbody>
</table>

- message queues are 1/4 the cost of UDP
  - less top-heavy
  - less data copying
# Ethernet Experiment

## Timing of Components of UDP Send

<table>
<thead>
<tr>
<th>Measurement Instant</th>
<th>Time (in ms)</th>
<th>Protocol Component</th>
<th>Duration (in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>after socket layers</td>
<td>1.74</td>
<td>socket layers</td>
<td>1.74</td>
</tr>
<tr>
<td>after connection</td>
<td>2.28</td>
<td>connection</td>
<td>0.54</td>
</tr>
<tr>
<td>end of UDP output</td>
<td>2.58</td>
<td>UDP send</td>
<td>0.30</td>
</tr>
<tr>
<td>before IP checksum</td>
<td>3.10</td>
<td>IP send</td>
<td>0.52</td>
</tr>
<tr>
<td>end of IP output</td>
<td>3.22</td>
<td>IP checksum</td>
<td>0.12</td>
</tr>
<tr>
<td>entire round-trip including receive</td>
<td>7.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Timing of Simple Ethernet Components

<table>
<thead>
<tr>
<th>Protocol Component</th>
<th>Time (in ms)</th>
<th>Measurement Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>user-level code</td>
<td>0.08</td>
<td>user</td>
</tr>
<tr>
<td>system call</td>
<td>0.33</td>
<td>send - user</td>
</tr>
<tr>
<td>device layering</td>
<td>0.61</td>
<td>null - send</td>
</tr>
<tr>
<td>kernel transmit</td>
<td>1.36</td>
<td>kping</td>
</tr>
<tr>
<td>mbufs on read</td>
<td>0.45</td>
<td>se_fast_return</td>
</tr>
<tr>
<td>context switch</td>
<td>0.40</td>
<td>insomnia - send</td>
</tr>
<tr>
<td>kernel-to-user copy</td>
<td>0.30</td>
<td>se_null_read</td>
</tr>
</tbody>
</table>
Evaluation of Unix IPC model (I)

Timing for UDP send

```
     Time (ms)
    /   5000  \
   /     4000 \
  /       3000 \n /         2000 \n/           1000 \n0            0   

Packet length (bytes)
```

- syscall
- socket
- udp
- ip
- ethernet
- device
- interrupt
Evaluation of Unix IPC model (II)

Timing for UDP receive

Time (ms)

Packet length (bytes)
The PUSH system

**Multicasting Timing (in ms)**

<table>
<thead>
<tr>
<th>Number of destinations</th>
<th>kernel level SE</th>
<th>user level SE</th>
<th>Push</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.2</td>
<td>1.2</td>
<td>2.7</td>
</tr>
<tr>
<td>5</td>
<td>4.2</td>
<td>5.9</td>
<td>6.6</td>
</tr>
<tr>
<td>10</td>
<td>8.0</td>
<td>11.7</td>
<td>11.0</td>
</tr>
<tr>
<td>15</td>
<td>11.7</td>
<td>17.5</td>
<td>15.6</td>
</tr>
<tr>
<td>20</td>
<td>15.4</td>
<td>23.4</td>
<td>20.2</td>
</tr>
</tbody>
</table>
Evaluation of Multicasting (I)

Approaches for Multicasting

(a) User-level SE multicast
(b) Kernel-level SE multicast
(c) Push multicast
Evaluation of Multicasting (II)

Multicasting cost

- User multicast
- Push multicast
- Kernel multicast
- Physical multicast

Time (ms)

Number of destinations
Raid Communication Subsystem, V.1

<table>
<thead>
<tr>
<th>High-level Raid communication (e.g., AD_StartCommit_RC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Data Representation (XDR)</td>
</tr>
<tr>
<td>Raid datagrams (e.g., SendPacket)</td>
</tr>
<tr>
<td>Long datagrams (e.g., sendto_ldg)</td>
</tr>
<tr>
<td>Datagram sockets (e.g., sendto)</td>
</tr>
<tr>
<td>UDP</td>
</tr>
<tr>
<td>IP</td>
</tr>
<tr>
<td>Ethernet</td>
</tr>
</tbody>
</table>

plus a separate process: name server oracle
Evaluation of Raid communication V.1

Raid servers' time

Legend:  AC = atomicity controller  
         CC = concurrency controller  
         AM = access manager  
         RC = replication controller  
         AD = action driver
RAID Elapsed Time for Transactions

<table>
<thead>
<tr>
<th>transaction</th>
<th>1 site</th>
<th>2 sites</th>
<th>3 sites</th>
<th>4 sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>select one tuples</td>
<td>0.3</td>
<td>0.3</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>select eleven tuples</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>insert twenty tuples</td>
<td>0.6</td>
<td>0.6</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>update one tuple</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
</tbody>
</table>
RAID Atomicity Control CPU Time

<table>
<thead>
<tr>
<th></th>
<th>1 site</th>
<th>2 sites</th>
<th>3 sites</th>
<th>4 sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>transaction</td>
<td>user</td>
<td>sys</td>
<td>user</td>
<td>sys</td>
</tr>
<tr>
<td>select one tuples</td>
<td>0.04</td>
<td>0.14</td>
<td>0.06</td>
<td>0.14</td>
</tr>
<tr>
<td>select eleven tuples</td>
<td>0.04</td>
<td>0.08</td>
<td>0.02</td>
<td>0.04</td>
</tr>
<tr>
<td>insert twenty tuples</td>
<td>0.20</td>
<td>0.16</td>
<td>0.08</td>
<td>0.14</td>
</tr>
<tr>
<td>update one tuple</td>
<td>0.04</td>
<td>0.10</td>
<td>0.06</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Times reported in seconds.
RAID Concurrency Control CPU Time

<table>
<thead>
<tr>
<th>server</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>transaction</td>
<td>user 0.04</td>
</tr>
<tr>
<td>select one tuple</td>
<td>0.02</td>
</tr>
<tr>
<td>select eleven tuples</td>
<td>0.12</td>
</tr>
<tr>
<td>insert twenty tuples</td>
<td>0.02</td>
</tr>
<tr>
<td>update one tuple</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Times reported in seconds.
Network Configurations for Raid

(a) LAN

(b) 19200 bps serial line

(c) Internetwork of LANs

Transaction Execution Time on Different Communication Topologies (in ms)

<table>
<thead>
<tr>
<th>Transaction</th>
<th>a-1</th>
<th>a-2</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>select one tuple</td>
<td>100</td>
<td>100</td>
<td>240</td>
<td>120</td>
</tr>
<tr>
<td>insert twenty tuples</td>
<td>320</td>
<td>380</td>
<td>520</td>
<td>400</td>
</tr>
<tr>
<td>update one tuple</td>
<td>100</td>
<td>120</td>
<td>260</td>
<td>120</td>
</tr>
</tbody>
</table>
Improvement Emphasis for Raid Communication subsystem Version 2

- Avoid top-heavy communication abstractions

- Reduce kernel interaction

- Minimize message copying

- Use a simple IPC memory management

- Adopt same mechanism for local and remote

- Exploit the nature of DTP
Raid Communication Subsystem V.2

- Port: shared by process and kernel

- Protocol: SE for LAN

- Naming: simple naming scheme:
  \( \langle \text{Raid instance, server, server instance} \rangle \Leftrightarrow \text{port number} \)

- Multicasting: each site sets a multicasting address using the transaction ID.

- Communication Primitives
Structure of Raidcomm V.2
Structure of a Communication Port

<table>
<thead>
<tr>
<th>trmlen</th>
<th>Transmission Buffer</th>
<th>Active Buffers</th>
</tr>
</thead>
<tbody>
<tr>
<td>len1</td>
<td>Receive Buffer 1</td>
<td></td>
</tr>
<tr>
<td>len2</td>
<td>Receive Buffer 2</td>
<td></td>
</tr>
<tr>
<td>len3</td>
<td>Receive Buffer 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>lenN</td>
<td>Receive Buffer N</td>
<td></td>
</tr>
</tbody>
</table>
Performance: Communication Primitives

- Goal: to evaluate the performance of Raid-comm Version 2

- Experiments: measure the local and remote round trip times
  - Add socket-based IPC and two SYS V local IPC methods for comparison

- Conclusions:
  - Our protocol is extremely lightweight
  - Most of the local round trip time is due to context switch overhead
Evaluation of Raidcomm V.2

Round-trip times

Time (μs)

Packet length (bytes)
Evaluation of Raidcomm V.2

Round-trip times

![Bar chart showing round-trip times for different components. The chart compares system times in Version 1 and Version 2.](chart.png)
Problem with XDR

- A data representation standard
- Doesn't take into account the high-level demands
- Encoding and decoding are expensive
- Unnecessary in most cases
Evaluation of XDR

Average user times for a transaction in a single machine

![Graph showing user and XDR times for different operations (AD, RC, AC, CC, AM) for two versions of Raidcomm (Version 1 and Version 2).]
Problem with context switch and scheduling

- OS scheduling policies do not consider high level relationships in a group of processes.

- Raid scenario:
  CC is CPU intensive and its priority is decreasing after some times. This forces CC is give up CPU after processing only one message, even though its time slice has not yet expired.

- context switches caused by synchronization are expensive.
Evaluation of Context Switch

The performance of context switch

- 2 semaphores P-V operation
- signal (SIGIO)
- explicit passing

Context switch round trip time (μs)
Improvement Emphasis for Raid Communication subsystem Version 3

- Avoid unnecessary data formatting
- Provide fast complex data transfer
- Eliminate kernel’s involvement in communication activities
- Use shared-memory as communication channel
- Reduce context switch overhead
- Use explicit control passing scheme
Raid Communication Subsystem V.3

- Improve interprocess communication inside one machine

- Port: shared by receiver, sender process and kernel

- Communication channel: pairwise shared memory segment

- Explicit control passing

- Communication Primitives
Evolution of Raid communication subsystems

• Version 1
  – based on UDP only
  – user-level multicasting

• Version 2
  – designed for LAN and local environment
  – use SE protocol and physical multicasting
  – mapping memory between kernel and process

• Version 3
  – improve local communication
  – support efficient complex data objects transfer
  – shared memory between processes
  – explicit control passing
Evaluation

The performance comparison of communication library

<table>
<thead>
<tr>
<th>Message</th>
<th>Length (bytes)</th>
<th>Raidcomm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V.1 (μs)</td>
<td>V.2 (μs)</td>
</tr>
<tr>
<td>SendNull († multicast dest = 5)</td>
<td>44</td>
<td>2462</td>
</tr>
<tr>
<td>MultiNull †</td>
<td>44</td>
<td>12180</td>
</tr>
<tr>
<td>SendTimestamp</td>
<td>48</td>
<td>2510</td>
</tr>
<tr>
<td>SendRelationDescriptor</td>
<td>76</td>
<td>2652</td>
</tr>
<tr>
<td>MultiRelationDescriptor †</td>
<td>72</td>
<td>12330</td>
</tr>
<tr>
<td>SendRelation</td>
<td>156</td>
<td>3864</td>
</tr>
<tr>
<td>SendWriteRelations</td>
<td>160</td>
<td>3930</td>
</tr>
</tbody>
</table>
Evaluation

Comparison of message round trip times

Time (µs)

Packet length (bytes)
Evaluation

User times for Raid servers
WANs Communication for DTP

- Approaches
- Problems
- Solutions
- Our experiences
General Approaches for WANs

• Dedicated links

• Bandwidth reservation (ATM network)

• Reliable transport service (by retransmission)
  
  – expensive

  – acknowledgement and retransmission may worsen the congestion problem

  – connection-mode is not scalable
Our approaches for WANs

- Clustering
  
  - A *cluster* is a set of sites that the link between each pair of members are reliable enough to use datagram protocols without error control.

  - Occasionally packet loss is remedied by timeout and rolling back the transaction to the previous checkpoint.

  - It reflects the network characteristics and architecture.

- Two-level communication scheme

- Surveillance
Communication performance of the Internet

- Objective:
  message delivery is essential to DTP.

- Measurements:
  round-trip time, message loss rate.

- Experimental method:
  the Internet echo services: IP/UDP/TCP echo.
Experiment I: Performance of WAN communication with time as variant

- Procedure:
  - from 1/08/92 to 2/05/92
  - between raid8.cs.purdue.edu and airmics.gatech.edu
  - one batch of 22 messages in every 5 minutes

- Data:
  MTFRT – Mean time for round trip
Time Variant

Hourly mean-time for round trip messages
Time Variant (cont’d)

Hourly loss rate for messages

Hourly Packets Loss Rate (Jan/08/1992 -- Feb/05/1992)

percentage of packages loss

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Jan/8 W Th F Sa Su M T W Th F Sa Su M T W Th F Sa Su M T W Th F Sa Su M T W
date of the week

Failure

packets loss ----- 0% line -----
Time Variant (cont’d)

Distribution of the hourly data

<table>
<thead>
<tr>
<th>mean-time for RT (ms)</th>
<th>number of hours</th>
<th>pct</th>
<th>average variance under such</th>
<th>average loss rate MTFRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 – 95</td>
<td>1</td>
<td>(0.1%)</td>
<td>2.08</td>
<td>2.3%</td>
</tr>
<tr>
<td>95 – 100</td>
<td>66</td>
<td>(9.8%)</td>
<td>0.74</td>
<td>2.7%</td>
</tr>
<tr>
<td>100 – 102</td>
<td>57</td>
<td>(8.5%)</td>
<td>0.94</td>
<td>2.1%</td>
</tr>
<tr>
<td>102 – 104</td>
<td>163</td>
<td>(24.3%)</td>
<td>1.02</td>
<td>3.6%</td>
</tr>
<tr>
<td>104 – 106</td>
<td>184</td>
<td>(27.4%)</td>
<td>1.27</td>
<td>3.4%</td>
</tr>
<tr>
<td>106 – 108</td>
<td>119</td>
<td>(17.7%)</td>
<td>1.60</td>
<td>4.2%</td>
</tr>
<tr>
<td>108 – 110</td>
<td>34</td>
<td>(5.1%)</td>
<td>2.40</td>
<td>4.3%</td>
</tr>
<tr>
<td>110 – 130</td>
<td>25</td>
<td>(3.7%)</td>
<td>3.97</td>
<td>5.9%</td>
</tr>
<tr>
<td>130 – 180</td>
<td>21</td>
<td>(3.1%)</td>
<td>14.52</td>
<td>7.8%</td>
</tr>
<tr>
<td>failure</td>
<td>2</td>
<td>(0.3%)</td>
<td>N/A</td>
<td>100.0%</td>
</tr>
<tr>
<td>total</td>
<td>672</td>
<td>(100.0%)</td>
<td>1.76</td>
<td>4.0%</td>
</tr>
</tbody>
</table>
Time Variant (cont'd)

Average round-trip time in normal work days

[Diagram showing round-trip time in milliseconds for different times of the day on Tuesdays, Wednesdays, and Thursdays.]
Time Variant (cont'd)

Average loss rate in normal work days

Packet Loss Rate in (Tue, Wed, Thu)

percentage of packets loss

12am 3am 6am 9am noon 3pm 6pm 9pm 12am

loss rate

time of the date
Observation for Time Variant

- There are large variations in parameters such as communication delay and messages loss.

- The variations exists in two dimensions: along the time axis and across the networks.

- The time of day has strong influence on the message delivery.
  - The message delay is slightly longer around noon.
  - Hourly peak phenomenon.
Experiment II: Performance of WAN communication with site as variant

• Procedure:
  
  – selected over 2000 major hosts from American universities and colleges.

  – collected data from 1843 hosts and 505 local networks

• Major hosts used:
  Mostly are backbone host such as mail homes. They are good representative for measurements.

• Measurement: round-trip time
Site Variant

Distribution of round-trip time to a host

The distribution of the RTT among 1843 major hosts

Examples:
160 sites out of the 2000 have the RTT at 80ms, 7 others at 300ms, etc.

Out of bound data:
range (ms)  num
1000-2000   20
2000-3000   5
3000-4000   2
Site Variant (cont'd)

Distribution of round-trip time to a network

The distribution of the RTT to 505 major local networks

Examples:
40 networks out of the 505 have the RTT at 90ms, 4 others at 300ms, etc.

Out of bound data:
9 networks have the RTT over 1000ms
Observation for site variant

• The message delivery performance is unbalanced across the networks.

• Most of the major hosts can be reached within 400ms round-trip time.

• There is "clustering" effect in the network topology; different sites in a same local network have similar behaviour in the viewpoint of an outside host.
Impact of WAN communication on DTP

- The time to delivery a transaction message is substantially longer. This implies that a transaction has to stay much longer in the system, which will increase the contention of a database system.

- The timeout mechanism is affected. The timeout value should be a function of the number of remote messages needed for the transaction as well as the messages' destination.

- Many of the local area network solution is not applicable or insignificant to wide area network.
  e.g. physical multicasting
Impact on DTP (cont’d)

- The focus on improving communication should be on reducing the number of messages generated by the system.

- More reliable transport service are needed.

- DTP algorithms need to take into account the highly varying in parameters such as message delay and loss.
  e.g. quorum consensus replication method

- Surveillance facilities that constantly monitor the network and collect performance data are helpful to improve the performance of DTP.
Communication Issues in DTP systems

- Reliability vs efficiency
  UDP is suitable for local area networks, but not reliable enough for wide area networks.

- Scalability
  Virtual circuit type of connection mode protocol (e.g. TCP) has scalability problem.

- Two-level communication scheme
  - clustering sites by the communication performance
  - inter-cluster communication: UDP
  - intra-cluster communication: TCP
Surveillance technique for improving DTP

To export the knowledge about WAN performance to the DTP knowledge that is relevant

<table>
<thead>
<tr>
<th>Layers</th>
<th>Metrics of information static</th>
<th>Metrics of information dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comm.</td>
<td>type of link</td>
<td>message delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>throughput</td>
</tr>
<tr>
<td></td>
<td></td>
<td>loss rate</td>
</tr>
<tr>
<td>OS</td>
<td>CPU speed</td>
<td>system load</td>
</tr>
<tr>
<td>DTP</td>
<td></td>
<td>response time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transaction load</td>
</tr>
</tbody>
</table>
Surveillance (cont'd)

Applications That Benefit

- Replication Control
  quorum selection (to select the minimal subset of sites that has the best predicted performance)

- Atomic Control
  commitment (to early abort the transactions that cannot continue due to changes in network condition)

- Adaptable Communication Subsystem
  (to change error recovery strategic according to the network dynamics)
Experimenting DTP in WAN

- Experimental Facility for WAN
  - General approaches
  - Our approach: emulation

- Performance of DTP in WAN
Experimental Facility for WAN

- General Approach 1: Real experiment
  - small scale, designated sites only
  - spans multiple administrative boundaries, had to control,
  - expensive and sometimes unrealistic

- General Approach 2: Simulation
  - highly scalable
  - need to simplify the model
  - need to justify the input parameters
  - need to collect a substantially large trace
Emulation Approach

- Any 2 hosts in a LAN emulate 2 hosts geographically separated over the world.

- Communication between these 2 hosts goes through the real path of the 2 emulated hosts.

- Justification: what makes the difference between DTP for LAN and for WAN?
  - not the actual location of sites (as long as they are the comparable models),
  - but the communication path.
WANCE tool

Software systems that implement the emulation scheme and allow the emulation experiments of arbitrary sites in the wide area network.
Performance of DTP in WAN

Purdue–Helsinki experiment

Response Time

Throughput

Abort Rate
Performance of DTP in WAN

Purdue–Hong Kong experiment

Response Time

Throughput

Abort Rate

(time of the day)
Performance of DTP in WAN

Elapsed time for basic operations

![Graph showing response time vs. number of tuples for different operations in a WAN network. The graph includes lines for update (Hong Kong), update (Helsinki), update (UTA), and update (LAN), as well as insert (Hong Kong), insert (Helsinki), insert (UTA), and insert (LAN). The select (LAN) operation is also depicted with a different marker.]
Performance of DTP in WAN

2PC vs 3PC

<table>
<thead>
<tr>
<th>configuration</th>
<th>response time</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2PC</td>
<td>3PC</td>
<td>overhead</td>
</tr>
<tr>
<td>within LAN</td>
<td>0.6</td>
<td>0.7</td>
<td>16%</td>
</tr>
<tr>
<td>Purdue–UTA</td>
<td>0.7</td>
<td>0.8</td>
<td>14%</td>
</tr>
<tr>
<td>Purdue–Helsinki</td>
<td>1.3</td>
<td>2.0</td>
<td>53%</td>
</tr>
<tr>
<td>Purdue–Hong Kong</td>
<td>3.5</td>
<td>5.5</td>
<td>57%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>configuration</th>
<th>time spent in AC</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2PC</td>
<td>3PC</td>
<td>overhead</td>
</tr>
<tr>
<td>within LAN</td>
<td>0.3</td>
<td>0.4</td>
<td>33%</td>
</tr>
<tr>
<td>Purdue–UTA</td>
<td>0.4</td>
<td>0.5</td>
<td>25%</td>
</tr>
<tr>
<td>Purdue–Helsinki</td>
<td>0.9</td>
<td>1.6</td>
<td>77%</td>
</tr>
<tr>
<td>Purdue–Hong Kong</td>
<td>2.8</td>
<td>4.8</td>
<td>71%</td>
</tr>
</tbody>
</table>
Results and Conclusions

- The basic communication services required to support distributed transaction processing are
  - multicasting
  - remote procedure call (RPC)
  - inexpensive datagram services
  - local interprocess communication (IPC)

- Identified the problem in general purpose IPC

- Experiment and evaluate various communication model through the evolution of Raid communication software
Results of Our Research

(continued)

- Showed how several ideas fulfill the needs of transaction processing
  - lightweight protocol
  - simple naming scheme
  - memory mapping and shared memory
  - physical multicasting
  - explicit control passing
  - increasing adaptability

- Developed transaction-oriented communication facilities
  - Reduces kernel overhead during transaction processing in Raid by up to 70%
  - Reduces user level overhead during communication in Raid by up to 30%
References


References (cont'd)


