Distributed Systems Research that Makes a Difference

Yair Amir

With: Baruch Awerbuch, R. Sean Borgstrom, Ryan Caudy, Claudiu Danilov, Ashima Munjal, Cristina Nita-Rotaru, Theo Schlossnagle, Jonathan Stanton, Ciprian Tutu

Center for Networking and Distributed Systems
Johns Hopkins University

www.cnnds.jhu.edu

Distributed Systems

• Distributed system is:
  – A collection of computers.
  – Connected by a network.
  – With a common goal.

• We at CNDS care about:
  – High availability.
  – High performance.
Hopkins/CNDS Research Objective

- **Understand** the problem: distributed communication and information access.
  - Availability, consistency, performance tradeoffs...
  - Local / wide area networks, latency, throughput...
- **Invent** correct and efficient algorithmic solutions to cope with the problematic aspects of distributed systems.
- **Develop** a small set of generic software tools encapsulating these solutions.
- **Use** these software tools to create a paradigm shift in the way distributed systems are actually built:
  - Through actual use of these software tools.
  - Through education.

Outline

- CNDS vision.
- Group communication.
  - What is it?
  - The Spread toolkit.
  - Distributed logging with Spread.
- Metacomputers.
  - Static and dynamic resource allocation.
  - The Cost-Benefit framework.
  - Backhand – load balancing a web cluster.
- Wackamole.
  - High availability for clusters.
  - High availability for edge routers.
- Making a difference.
Group Communication:  
A different communication paradigm

- Handles potential problems in the network: message loss, machine downtime, network partitions, security threats.
- Delivery guarantees: Unreliable, Reliable, Safe (stable).
- Message ordering: Unordered, FIFO, Agreed order.
- Membership services.
- Group security: Unsecure, Signed, Encrypted.

Yair Amir 10 Dec 02
What Can You Do With Group Communication?

• Efficient server replication
  – High throughput.
  – High availability.

• And also:
  – Distributed system management.
  – Conferencing.
  – Collaborative Design.
  – Distributed simulations.
  – …

Spread: A Group Communication Toolkit

• Client – Daemon architecture.
• Very simple API (basically 6 calls). C/C++, Java, Perl, Python…
• Cross platform. Windows, Solaris, Alpha, MacOS, Embedded…
• Part of OpenLinux, Debian, FreeBSD.

Yair Amir 10 Dec 02
A Spread Overlay Network

- Uses membership knowledge to optimize performance.

Distributed Logging

- Think of a cluster of 50 (web) servers.
- Operation logs for all of the servers should be consolidated in one logical place.
- Aggregate log should be resilient (so it must be replicated).
Distributed Logging (cont.)

Today, mainly a service for local area clusters.

Useful Spread properties:

• Scalability with the number of groups.
• Open group semantics - good support for publish-subscribe.
• Membership services.
• Efficient for small messages.
• Small footprint - low overhead.

Implementation

• Option 1: Distributed syslogd
  – Adding a Spread group as a potential destination of the operating system logger syslogd (in addition to file, screen, socket).
• Option 2: mod_log_spread
  – An Apache module enabling logging through Spread (courtesy of George Schlossnagle).
The Real Challenge

- Surprisingly, not the basic protocols.
- But:
  - **Hostile** environment - average load on the servers may top **100**!
  - They keep adding these servers...
  - Has to be rock-solid, running constantly for days.

Status at bp12 V3.12 (state 1, gstate 1) after **81113** seconds:

- Membership: **34** procs in **1** segments, leader is **bp12**
- rounds: **2796352**
- tok_hurry: **8537**
- memb change: **5**
- sent pack: **625516**
- recv pack: **27229735**
- retrans: **553**
- u retrans: **460**
- s retrans: **93**
- b retrans: **0**
- My_aru: **21829811**
- Aru: **21829718**
- Highest seq: **21829811**
- Sessions: **67**
- Groups: **2**
- Window: **60**
- Deliver M: **27552100**
- Deliver Pk: **27855251**
- Pers Window: **15**
- Delta Mes: **8076**
- Delta Pack: **8166**
- Delta sec: **10**

Distributed Logging (cont.)

- Once you have it running, there is more:
  - **Real-time monitoring.**
  - **Real-time data-mining and customization.**
Outline

• CNDS vision.
• Group communication.
  – What is it?
  – The Spread toolkit.
  – Distributed logging with Spread.
• Metacomputers.
  – Static and dynamic resource allocation.
  – Backhand – load balancing a web cluster.
• Wackamole.
  – High availability for clusters.
  – High availability for edge routers.
• Making a difference.

Static and Dynamic Distributed Resource Allocation

We looked at two metacomputing systems: PVM and Mosix.

• PVM - Parallel Virtual Machine
  – Static Assignment of jobs to machines.
  – Default - round robin assignment policy.
  – Widely used.
• Mosix
  – Dynamic process migration.
  – Main objective is load balancing, with some ad-hoc heuristics for memory depletion.
A Cost Benefit Approach to Distributed Resource Allocation

Resource A (CPU)

Resource B (memory)

Resource C (I/O)

Machine_cost = cost(CPU) + cost(memory) + cost(I/O)

New Assignment Policy

Based on the Cost Benefit framework, We have created two additional policies:

- Enhanced PVM (EPVM)
- Enhanced Mosix (Emosix).

We compared the performance of the four policies according to the average slowdown criterion.
Simulation: PVM / EPVM

Average Slowdown (lower is better)

Yair Amir
10 Dec 02

Simulation: Mosix / EMosix

Average Slowdown (lower is better)

Yair Amir
10 Dec 02
Real Life: PVM / EPVM

Average Slowdown (lower is better)

Interesting: Mosix / EPVM

Average Slowdown (lower is better)
How general is the framework?
Managing a Web farm

• Applying the same concept to manage the resources of a Web farm.
• Web servers in the farm may have different capabilities. Tasks may have different requirements.
• Hard Problems:
  – Relatively short lived tasks.
  – Stale information (welcome to the real world).
  – Inaccurate information (welcome to the real world - did we say that?).

Yair Amir 10 Dec 02

Backhand:
Load Balancing A Web Cluster

• Peer-to-peer, cost-based resource allocation decisions.
• The necessary “machinery” implemented as an Apache module to minimize overhead.
• Linux, Solaris, FreeBSD, Windows*
• Part of SuSE Linux and Open Linux.
• www.backhand.org

Yair Amir 10 Dec 02
Backhand in Action

Yair Amir 10 Dec 02 25

Wackamole

Yair Amir 10 Dec 02 26
Wackamole
Outline

• CNDS vision.
• Group communication.
  – What is it?
  – The Spread toolkit.
  – Distributed logging with Spread.
• Metacomputers.
  – Static and dynamic resource allocation.
  – Backhand – load balancing a web cluster.
• Wackamole.
  – High availability for clusters.
  – High availability for edge routers.
• Making a difference.

Wackamole - Dynamic Cluster with N-way Fail-over

• In contrast to available gateway solutions (Alteon, BigIP, Cisco’s Local-director, …).
• Exploiting Spread’s strong membership semantics to build a distributed agreement protocol that strictly distributes the cluster’s public IP addresses between the currently connected servers.
• Useful for locally replicated web servers, ftp servers, DNS servers, and even fail-over routers.
• Transparent!
  www.wackamole.org
Wackamole Architecture

System 1
- Wackamole
  - State Synchronization Algorithm
  - IP Address Manager
  - Group Communication System

System 2
- Wackamole
  - State Synchronization Algorithm
  - IP Address Manager

Network

IP Addresses
- System 1:
  - xxx.xxx.221.151
  - xxx.xxx.221.152
  - xxx.xxx.221.153

- System 2:
  - xxx.xxx.221.154
  - xxx.xxx.221.156

Wackamole for Clusters

Internet

Router
- xxxx.xxx.221.1

Webserver 1
- xxxx.xxx.221.151

Webserver 2
- xxxx.xxx.221.152

Webserver 3
- xxxx.xxx.221.153

Webserver 1
- xxxx.xxx.221.151
- xxxx.xxx.221.153

Webserver 2
- xxxx.xxx.221.152

Webserver 3

Wackamole: Atp-Spoof

Wackamole: Reallocate IP

Yair Amir
10 Dec 02
Wackamole for Edge Routers

Wackamole Performance

- Out of the box average for fail-over: 12 seconds.
- Tuned average for fail-over: 2.5 seconds.
- Voluntary fail-over (maintenance, etc): up to 250 milliseconds, most of the time within 10 milliseconds.
Making a Difference

10,000 actual sites discovered

100 actual sites discovered

Backhand-powered web sites discovery thanks to Netcraft

Yair Amir

10 Dec 02