Affinity in Distributed Systems

Thesis defense

Ymir Vigfusson

Joint work with:
Hussam Abu-Libdeh, Mahesh Balakrishnan, Ken Birman, Gregory Chockler, Qi Huang, Jure Leskovec, Deepak Nataraj and Yoav Tock.
Group communication

• Most network traffic is *unicast* communication (one-to-one).

• But a lot of content is identical:
  – Audio streams, video broadcasts, system updates, *etc.*

• To minimize redundancy, would be nice to *multicast* communication (one-to-many).
Multicast by Unicast
Gossip
<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Deliv. Speed</th>
<th>Redundancy</th>
<th>Scalable in # users?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-to-point unicast</td>
<td>Slow</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>IP Multicast (IPMC)</td>
<td>Fast</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Gossip</td>
<td>Slow</td>
<td>Low</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Talk Outline

• Dr. Multicast (MCMD)
  – Group scalability in IP Multicast.

• Gossip Objects (GO) platform
  – Group scalability in gossip.

• Affinity
  – GO+MCMD optimizations based on group overlaps
    – Explore the properties of overlaps in data sets

• Conclusion
IP Multicast in Data Centers

- Smaller scale – well defined hierarchy
- Single **administrative** domain
- Firewalled – can ignore malicious behavior
IP Multicast in Data Centers

• Useful, but rarely used.
• Various problems:
  – Security
  – Stability
  – Scalability
IP Multicast in Data Centers

![Graph showing packet miss rate (%) against number of groups joined]
IP Multicast in Data Centers

• Useful, but rarely used.
• Various problems:
  – Security
  – Stability
  – Scalability
• **Bottom line:** Administrators have no *control* over IPMC.
  – Thus they choose to disable it.
Wishlist

• **Policy:** Enable control of IPMC.

• **Transparency:** Should be backward compatible with hardware and software.

• **Scalability:** Needs to scale in number of groups.

• **Robustness:** Solution should not bring in new problems.
Acceptable Use Policy

• Assume a higher-level network management tool compiles policy into primitives.

• Explicitly allow a process (user) to use IPMC groups.
  • allow-join(process ID, logical group ID)
  • allow-send(process ID, logical group ID)

• Point-to-point unicast always permitted.

• Additional restraints.
  • max-groups(process ID, limit)
  • force-udp(process ID, logical group ID)
Dr. Multicast (MCMD)

- Translates *logical* IPMC groups into either *physical* IPMC groups or multicast by unicast.
- Optimizes resource use.
Network Overhead

- Gossip Layer uses constant background bandwidth on average

2.1 kb/s
Application Overhead

- Insignificant overhead when mapping logical IPMC group to physical IPMC group.
Optimization questions

Users <-> Groups

BLACK

Users <-> Groups

Multicast
Optimization Questions

- Assign IPMC and unicast addresses s.t.
  - Min. receiver filtering
  - Min. network traffic
  - Min. # IPMC addresses
  - ... yet have all messages delivered to interested parties
Optimization Questions

- Assign IPMC and unicast addresses s.t.
  - $\leq \alpha \%$ receiver filtering (hard)
  - (1) Min. network traffic
  - $\leq M$ # IPMC addresses (hard)

- Prefers sender load over receiver load.

- Control knobs part of administrative policy.
MCMD Heuristic

Groups in `user-interest' space

(0,1,1,1,1,1,0,1,0,0,0,0,1,1)
MCMD Heuristic

Groups in `user-interest' space

224.1.2.3

224.1.2.4

224.1.2.5

Groups in `user-interest' space
MCMD Heuristic

Groups in `user-interest' space

Sending cost: [Diagram showing a low cost]

Filtering cost: [Diagram showing a high cost]
MCMD Heuristic

Groups in `user-interest' space

Sending cost: Unicast

Filtering cost: MAX
MCMD Heuristic

Groups in `user-interest' space

Unicast

224.1.2.3

224.1.2.4

224.1.2.5

Groups in `user-interest' space
Dr. Multicast

- **Policy:** Permits data center operators to selectively enable and control IPMC.

- **Transparency:** Standard IPMC interface to user, standard IGMP interface to network.

- **Scalability:** Uses IPMC when possible, otherwise point-to-point unicast.

- **Robustness:** Distributed, fault-tolerant service.
Talk Outline

• Dr. Multicast (MCMD)
  – Group scalability in IP Multicast.

• Gossip Objects (GO) platform
  – Group scalability in gossip.

• Affinity
  – GO+MCMD optimizations based on group overlaps
  – Explore the properties of overlaps in data sets

• Conclusion
Gossip

• **Def:** *Exchange information with a random node once per round.*

• Has appealing properties:
  – Bounded network traffic.
  – Scalable in group size.
  – Robust against failures.
  – Simple to code.

• When # of groups scales up, lose
GO Platform

Diagram showing the GO Platform with components such as Node, Rumor Queue, Membership Component, and multiple Applications (Apps). The GO Platform has a Gossip Mechanism, Event Loop, and GO Heuristic.
Random gossip

• **Recipient selection:**
  – Pick node $d$ uniformly at random.

• **Content selection:**
  – Pick a rumor $r$ uniformly at random.
Observations

• Gossip rumors usually small:
  – Incremental updates.
  – Few bytes hash of actual information.

• Packet size below MTU irrelevant.
  – Stack rumors in a packet.
  – But which ones?

• Rumors can be delivered indirectly.
  – Uninterested node might forward
Random gossip w. stacking

• **Recipient selection:**
  - Pick node $d$ uniformly at random.

• **Content selection:**
  - Fill packet with rumors picked uniformly at random.
GO Heuristic

• **Recipient selection:**
  – Pick node $d$ biased towards higher group traffic.

• **Content selection:**
  – Compute the *utility* of including rumor $r$
    • Probability of $r$ infecting an uninfected host when it reaches the target group.
  – Pick rumors to fill packet with probability proportional to utility.
• Recipient selection:
  – Pick node biased towards higher group traffic.

• Content selection:
  – Compute the utility of including rumor $r$.
  – Pick rumors to fill packet with probability proportional to utility.

• Probability of $r$ infecting an uninfected host when it reaches the target group.
Evaluation

- IBM Websphere trace (1364 groups)
Evaluation

- IBM Websphere trace (1364 groups)
Evaluation

- IBM Websphere trace (1364 groups)
Talk Outline

• Dr. Multicast (MCMD)
  – Group scalability in IP Multicast.

• Gossip Objects (GO) platform
  – Group scalability in gossip.

• Affinity
  – GO+MCMD optimizations based on group overlaps.
  – Explore the properties of overlaps in data sets.

• Conclusion
Affinity

• Both MCMD and GO have optimizations that depend on pairwise group overlaps (affinity).

• What degree of affinity should we expect to arise in the real-world?
Data sets/models

• What’s in a “group”? 
• Social:
  – Yahoo! Groups 
  – Amazon Recommendations 
  – Wikipedia Edits 
  – LiveJournal Communities 
  – *Mutual Interest Model*
• Systems:
  – IBM Websphere 
  – *Hierarchy Model*
• User and group degree distributions appear to follow power laws.

• Power-law degree distributions are often modeled by preferential attachment.

• Mutual Interest model:
  – Preferential attachment for bipartite graphs.
Systems Data Set

• IBM Websphere has remarkable structure!

• Typical for real-world systems?
  – Only one data point.
Systems Data Set

• Distributed systems tend to be hierarchically structured.

• **Hierarchy model**
  – Motivated by Live Objects.

**Thm:** Expect a pair of users to overlap in

\[ \Theta \left( \frac{n}{\log^2 n} \right) \]

groups.
Data sets/models

• Social:
  – Yahoo! Groups
  – Amazon Recommendations
  – Wikipedia Edits
  – LiveJournal Communities
  – Mutual Interest Model

• Systems:
  – IBM Websphere
  – Hierarchy Model
• **Def:** Similarity of groups $j,j'$ is $\text{SIM}(j, j') = \frac{|G_j \cap G_{j'}|}{\max\{|G_j|, |G_{j'}|\}}$.  

Wikipedia

LiveJournal
Group similarity

- **Def:** Similarity of groups \( j, j' \) is

\[
\text{SIM}(j, j') = \frac{|G_j \cap G_{j'}|}{\max\{|G_j|, |G_{j'}|\}}.
\]
Group similarity

- **Def:** Similarity of groups \( j, j' \) is 
  \[
  \text{SIM}(j, j') = \frac{|G_j \cap G_{j'}|}{\max\{|G_j|, |G_{j'}|\}}.
  \]
Baseline overlap

- Is the similarity we see a real effect?
- Consider a random graph with the same degree distributions as a baseline.
- **Spokes model:**

![Spokes model diagram]
Baseline overlap

- Plot difference between data and Spokes
  \[ \Delta(j, j') = \text{SIM}_\Gamma(j, j') - \text{SIM}_{\Gamma'}(j, j') \].
- At most 50 samples per group size pair.

<table>
<thead>
<tr>
<th>Data set/model</th>
<th>Avg. ( \Delta ) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wikipedia</td>
<td>-0.004</td>
</tr>
<tr>
<td>Amazon</td>
<td>0.031</td>
</tr>
<tr>
<td>Yahoo! Groups</td>
<td>0.000</td>
</tr>
<tr>
<td>Mutual Interest Model</td>
<td>0.006</td>
</tr>
<tr>
<td>IBM Websphere</td>
<td>0.284</td>
</tr>
<tr>
<td>Hierarchy Model</td>
<td>0.358</td>
</tr>
</tbody>
</table>

Looking pretty random
Conclusions

• Group communication important, but group scalability is lacking.

• **Dr. Multicast** harnesses IPMC in data centers.
  – **Impact:** HotNets paper + NSDI Best Poster award.
  – Solution being adopted by CISCO and IBM.
Conclusions

• **GO** provides group scalability for gossip.
  – **Impact:** LADIS paper + Invited to the P2P Conference.
  – Platform will run under the Live Objects framework.

• Characterizing and exploiting group affinity in systems is exciting current and future work.
## Publications

<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Conferences/Proceedings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GO: Platform Support For Gossip Applications.</strong></td>
<td>With Ken Birman, Qi Huang, Deepak Nataraj.</td>
<td>LADIS ‘09. Invited to P2P ’09.</td>
</tr>
<tr>
<td><strong>Adaptively Parallelizing Distributed Range Queries.</strong></td>
<td>With Adam Silberstein, Brian Cooper, Rodrigo Fonseca.</td>
<td>VLDB ’09.</td>
</tr>
<tr>
<td><strong>Dr. Multicast: Rx for Data Center Communication Scalability.</strong></td>
<td>With Hussam Abu-Libdeh, Mahesh Balakrishnan, Ken Birman, Yoav Tock.</td>
<td>Hotnets ‘08. LADIS ‘08. NSDI ‘08 (Best Poster).</td>
</tr>
<tr>
<td><strong>Hyperspaces for Object Clustering and Approximate Matching in P2P Overlays.</strong></td>
<td>With Bernard Wong, Emin Gun Sirer.</td>
<td>HotOS ‘07.</td>
</tr>
</tbody>
</table>
Baseline overlap

- Plot difference between data and *Spokes*
  \[ \Delta(j, j') = \text{SIM}_\Gamma(j, j') - \text{SIM}_{\ell}(j, j'). \]
- **Cell:** Avg. \( \Delta \) over particular group sizes.
Baseline overlap

- Plot difference between data and Spokes
  \[ \Delta(j, j') = \text{SIM}_G(j, j') - \text{SIM}_F(j, j'). \]
- **Cell**: Avg. \( \Delta \) over particular group sizes.

Websphere
Affinity results

- Social affinity pretty random.
- Websphere has substantial overlaps.
- MCMD Heuristic does well in all cases:
Conclusions

• Group communication important, but group scalability is lacking.

• Dr. Multicast harnesses IPMC in data centers.
  – **Impact**: HotNets paper + NSDI Best Poster award.
  – Solution being adopted by CISCO and IBM.

• **GO** provides group scalability for gossip.
  – **Impact**: LADIS paper + Invited to the P2P Conference.
  – Platform will run under the Live Objects framework.

• Characterizing and exploiting group affinity in systems is exciting current and future work.
Publications

• **GO: Platform Support For Gossip Applications.**

• **Adaptively Parallelizing Distributed Range Queries.**
  With Adam Silberstein, Brian Cooper, Rodrigo Fonseca. *VLDB* ‘09.

• **Slicing Distributed Systems.**

• **Dr. Multicast: Rx for Datacenter Communication Scalability.**

• **Hyperspaces for Object Clustering and Approximate Matching in P2P Overlays.**
  With Bernard Wong, Emin Gun Sirer. *HotOS* ‘07.