Balancing Gossip Exchanges in Networks with Firewalls

J. Leitão, R. van Renesse and L. Rodrigues

IPTPS 2010
April 27, 2010
1 Introduction

2 Balancing Gossip

3 Evaluation

4 Conclusions
Introduction

Scope

- Gossip protocols:
  - Very flexible.
  - Easy to implement.
  - Scalable.
Introduction

Balancing Gossip Exchanges in Networks with Firewalls

J. Leitão, R. van Renesse and L. Rodrigues

Introduction

Gossip protocols
Introduction
Gossip protocols
Introduction

Gossip protocols
Introduction
Gossip protocols
Introduction
Gossip protocols

State Transfer
Introduction

Gossip protocols
Introduction

Gossip Protocols.

Inherent load-balancing properties

Every participant will engage in a similar number of gossip exchanges.

Load Balancing...

Only true if considering a “flat” topology.
Inherent load-balancing properties

Every participant will engage in a similar number of gossip exchanges.

Load Balancing...

Only true if considering a "flat" topology.
Introduction
Introducing Firewalls and NAT boxes
Introduction

Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction

Introducing Firewalls and NAT boxes
Balancing Gossip Exchanges in Networks with Firewalls

J. Leitão, R. van Renesse and L. Rodrigues

Introduction

Introducing Firewalls and NAT boxes
Introduction

Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Introducing Firewalls and NAT boxes
Introduction
Confinement Domain & Unconfined Nodes
Introduction
Confinement Domain & Unconfined Nodes
Introduction

Confinement Domain & Unconfined Nodes

Balancing Gossip Exchanges in Networks with Firewalls
J. Leitão, R. van Renesse and L. Rodrigues

Introduction

Evaluation

Conclusions
Introduction
Confinement Domain & Unconfined Nodes
Introduction
Confinement Domain & Unconfined Nodes
Introduction
Confinement Domain & Unconfined Nodes
Introduction
Confinement Domain & Unconfined Nodes
Introduction
Confinement Domain & Unconfined Nodes
This can unbalance the system behavior:

- Unconfined nodes can participate in a much higher number of gossip exchanges.
- Specially when only a small fraction of nodes are unconfined.

This unbalance is undesirable:

- State reconciliation can require significant CPU Resources:
  - Techniques to reduce the use of bandwidth.
  - Encryption/decryption and signature/verification of messages.
  - Serialization/deserialization of objects.
Introduction

Motivation

This can unbalance the system behavior:

- Unconfined nodes can participate in a much higher number of gossip exchanges.
- Specially when only a small fraction of nodes are unconfined.

This unbalance is undesirable:

- State reconciliation can require significant CPU Resources:
  - Techniques to reduce the use of bandwidth.
  - Encryption/decryption and signature/verification of messages.
  - Serialization/deserialization of objects.
In this paper:

- We present a new approach to balance gossip exchanges in networks with firewalls.
  - only requires local information.
  - no coordination overhead.
  - nodes are not required to know if they are unconfined or confined.
Introduction

Contribution

In this paper:

- We present a new approach to balance gossip exchanges in networks with firewalls.
  - only requires local information.
  - no coordination overhead.
  - nodes are not required to know if they are unconfined or confined.
Outline

1 Introduction

2 Balancing Gossip
   - Rationale
   - Intuition
   - Example

3 Evaluation

4 Conclusions
We follow 2 observations.

Observation 1:
Two nodes in distinct confinement domains can only exchange information through an unconfined node.

Observation 2:
In a balanced system on average:
For each gossip exchange initiated by a node (on average) that node participates in a gossip exchange initiated by another peer.
Balancing Gossip
Rational

We follow 2 observations.

Observation 1:
Two nodes in distinct confinement domains can only exchange information through an unconfined node.

Observation 2:
In a balanced system on average:
For each gossip exchange initiated by a node (on average) that node participates in a gossip exchange initiated by another peer.
We follow 2 observations.

**Observation 1:**
Two nodes in distinct confinement domains can only exchange information through an unconfined node.

**Observation 2:**
In a balanced system on average:
For each gossip exchange initiated by a node (on average) that node participates in a gossip exchange initiated by another peer.
Balancing Gossip

Intuition

- Each node maintains:
  - A quota value (initially with a value of 1).
  - A single-entry cache for connections created by other nodes.

Every node in the system executes the same protocol.
Balancing Gossip

Intuition

- Each node maintains:
  - A quota value (initially with a value of 1).
  - A single-entry cache for connections created by other nodes.

Every node in the system executes the same protocol.
Balancing Gossip

Intuition

- The quota limits the number of gossip exchanges initiated by other peers that a node can accept.
- Nodes increase their quota when they initiate a gossip exchange.
- The connection cache keeps alive the last connection used by another peer to initiate a gossip exchange.
- When a node receives a gossip request and does not have a quota value above zero it forwards the request through the cached connection.
The quota limits the number of gossip exchanges initiated by other peers that a node can accept.

Nodes increase their quota when they initiate a gossip exchange.

The connection cache keeps alive the last connection used by another peer to initiate a gossip exchange.

When a node receives a gossip request and does not have a quota value above zero it forwards the request through the cached connection.
Intuition

- The quota limits the number of gossip exchanges initiated by other peers that a node can accept.
- Nodes increase their quota when they initiate a gossip exchange.
- The connection cache keeps alive the last connection used by another peer to initiate a gossip exchange.
- When a node receives a gossip request and does not have a quota value above zero it forwards the request through the cached connection.
Balancing Gossip

Example
Balancing Gossip

Example

A

1

U

1

B

1

1

C
Balancing Gossip

Example

Introduction

Balancing Gossip

Evaluation

Conclusions
Balancing Gossip

Example
Balancing Gossip

Example
Balancing Gossip

Example
Balancing Gossip

Example
Balancing Gossip

Example
Balancing Gossip

Example
Balancing Gossip

Example

Balancing Gossip
Exchanges in Networks with Firewalls
J. Leitão, R. van Renesse and L. Rodrigues

Introduction
Balancing Gossip
Evaluation
Conclusions
Balancing Gossip

Example

A

B

U

0

2

1

C

Introduction
Balancing Gossip
Evaluation
Conclusions
Some additional aspects:

- A gossip requests are forwarded a limited number of times (TTL).
- If a node has an empty connection cache it engages in the gossip exchange.
Outline

1. Introduction
2. Balancing Gossip
3. Evaluation
   - Experimental Setting
   - Experimental Results
4. Conclusions
Evaluation

Experimental Setting

- We conducted simulations in the Peersim simulator.
  - System composed of 12,800 nodes.
  - Distributed in a variable number of confinement domains:
    - From 1 (flat topology) to 12,100 (star topology).
  - Each communication step has a latency selected uniformly at random between 2 and 7.
Evaluation
Experimental Setting

- We conducted simulations in the Peersim simulator.
- System composed of 12,800 nodes.
- Distributed in a variable number of confinement domains:
  - From 1 (flat topology) to 12,100 (star topology).
- Each communication step has a latency selected uniformly at random between 2 and 7.
Evaluation
Experimental Setting

- We conducted simulations in the Peersim simulator.
- System composed of 12,800 nodes.
- Distributed in a variable number of confinement domains:
  - From 1 (flat topology) to 12,100 (star topology).
- Each communication step has a latency selected uniformly at random between 2 and 7.
Evaluation

Experimental Setting

Application

- Simple anti-entropy protocol.
  - All nodes have a state values initially set to 0.
  - A random node changes its state value to 1.
  - Nodes gossip their state value and update theirs with highest value.

- Each node initiates 500 gossip exchanges.
- If the system is balanced each node should participate in 1000 gossip exchanges.
Evaluation
Experimental Setting

Application

- Simple anti-entropy protocol.
  - All nodes have a state values initially set to 0.
  - A random node changes its state value to 1.
  - Nodes gossip their state value and update theirs with highest value.
- Each node initiates 500 gossip exchanges.
- If the system is balanced each node should participate in 1000 gossip exchanges.
Evaluation

Experimental Setting

- We evaluate our protocol using distinct TTL values:
  - TTL = 1 - Equivalent to regular gossip.
  - TTL = 2 - Each gossip request can be redirected one time.
  - TTL = 5.
  - TTL = 10.
Evaluation
Experimental Setting

- We evaluate our protocol using distinct TTL values:
  - TTL = 1 - Equivalent to regular gossip.
  - TTL = 2 - Each gossip request can be redirected one time.
  - TTL = 5.
  - TTL = 10.
Evaluation

Results show:

- Maximum latency (time until all nodes update their state value to 1).
- Maximum gossip exchanges performed by a single node.
- Maximum number of messages forwarded by a single node.
Evaluation

Results show:

- Maximum latency (time until all nodes update their state value to 1).
- Maximum gossip exchanges performed by a single node.
- Maximum number of messages forwarded by a single node.
Evaluation

Results

Results show:

- Maximum latency (time until all nodes update their state value to 1).
- Maximum gossip exchanges performed by a single node.
- Maximum number of messages forwarded by a single node.
Evaluation

Experimental Results: Maximum latency

- TTL = 1
- TTL = 2
- TTL = 5
- TTL = 10

Latency (time units) vs. number of confinement domains.

- 1
- 1600
- 3100
- 4600
- 6100
- 7600
- 9100
- 10600
- 12100
Evaluation
Experimental Results: Maximum gossip exchanges per node

- TTL = 1
- TTL = 2
- TTL = 5
- TTL = 10

number of confinement domains

number of gossip exchanges
Evaluation

Experimental Results: Maximum forwarded messages per node

- TTL = 2
- TTL = 5
- TTL = 10

Number of confinement domains: 1, 1600, 3100, 4600, 6100, 7600, 9100, 10600, 12100

Messages forwarded: 0, 10000, 20000, 30000, 40000, 50000

Bar chart showing the comparison of maximum forwarded messages per node for different TTL values with varying number of confinement domains.
Introduction

Balancing Gossip

Evaluation

Conclusions
Conclusions

- We have studied how to balance gossip exchanges in networks with firewalls.
- We proposed a new solution:
  - Effectively balances gossip exchanges.
  - Does not require nodes to know if they are confined or unconfined.
  - Has no coordination overhead.
- This technique can be easily implemented in current gossip-based mechanisms.
Conclusions

We have studied how to balance gossip exchanges in networks with firewalls.

We proposed a new solution:
- Effectively balances gossip exchanges.
- Does not require nodes to know if they are confined or unconfined.
- Has no coordination overhead.

This technique can be easily implemented in current gossip-based mechanisms.
Thanks.