

Implementation of a Large-Scale Epidemic Total Order Algorithm

Master Thesis

Jocelyn Thode¹ Hugues Mercier² Miguel Matos³

Neuchâtel, Switzerland, February 8, 2017

¹Department of Informatics, Université de Fribourg, Switzerland
jocelyn.thode@unifr.ch

²Institute of Computer Science, Université de Neuchâtel, Switzerland
hugues.mercier@unine.ch

³INESC-ID/IST, Universidade de Lisboa, Portugal
miguelmatos@di.uminho.pt

- Motivation
- EPTO explained
- LSDSUITE
- Evaluation
- Conclusion

Introduction

EPTO was only evaluated using a **simulation**. We need an evaluation with real peers to:

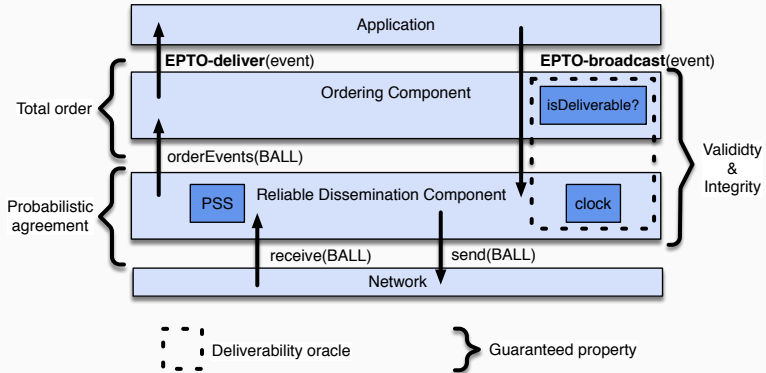
- Expose possible **limitations**
- Confirm simulation **results**

- Comparing EPTO meant testing it against other algorithms
- No framework to easily benchmark algorithms without having to rewrite them

Epidemic Total Order Algorithm:

- Probabilistic dissemination algorithm
 - Using balls-and-bins
- Provides deterministic total order, integrity and validity
- Scales well with the number of peers
 - Parameters increase logarithmically
- Churn resistant

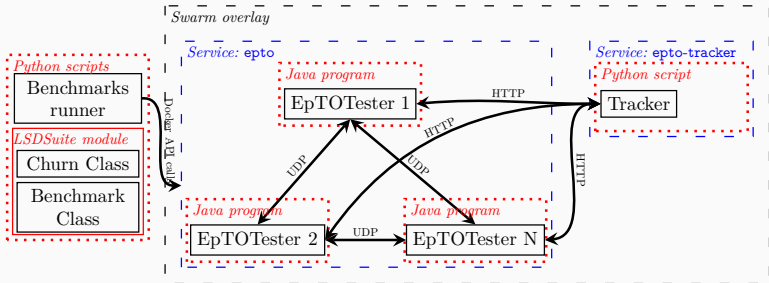
EPTO architecture



M. Matos, H. Mercier, P. Felber, R. Oliveira and J. Pereira, "EPTO: An epidemic total order algorithm for Large-scale distributed systems", in Proceedings of the 16th Annual Middleware Conference, ACM, 2015, pp. 100–111.

- Compatible with any distributed algorithm provided it runs on Docker
- support for a user-provided tracker
- Automated benchmarking execution
- Containers allow for more than 1 peer per physical node
- Can simulate churn or follow real traces

LSDSUITE architecture



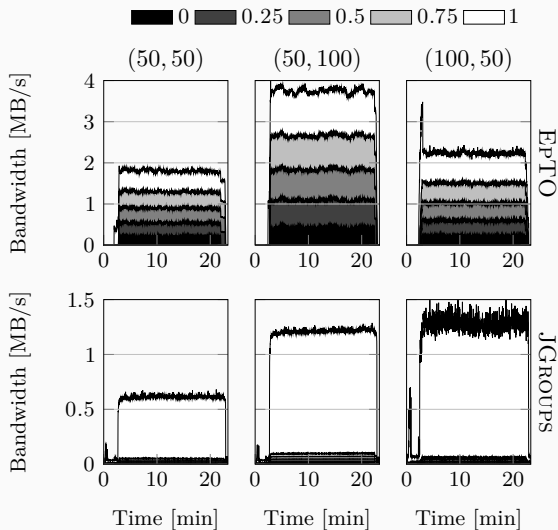
The protocol, churn and framework configuration is done through YAML files

The protocols logs must be written in a file to be extracted to the host

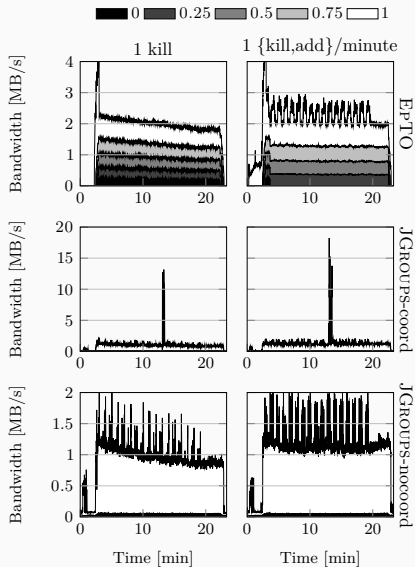
Evaluation

We evaluate EPTO against JGROUPS SEQUENCER, scaling peers and global event throughput per second.

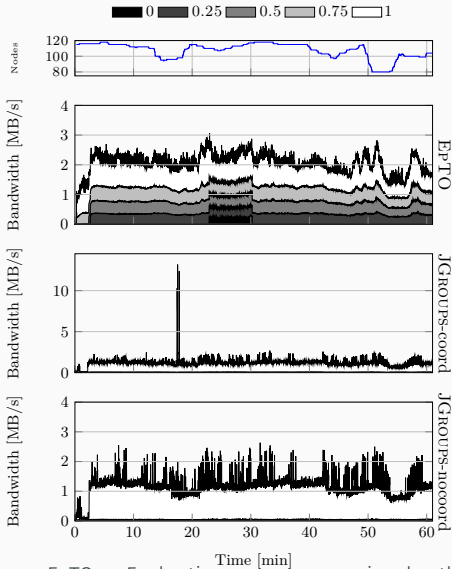
We write (n, e) where n is the number of peers and e is the global event throughput per second.



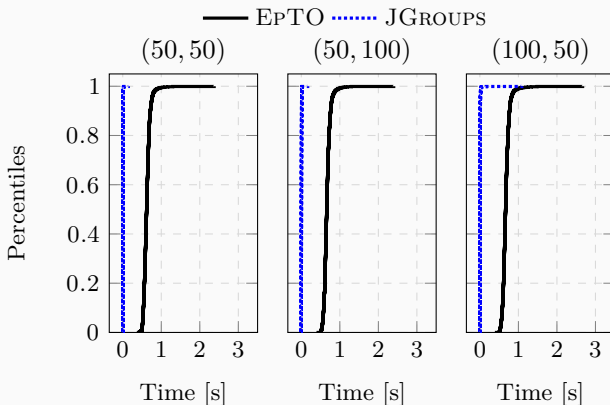
Bandwidth Synthetic Churn (100, 50)



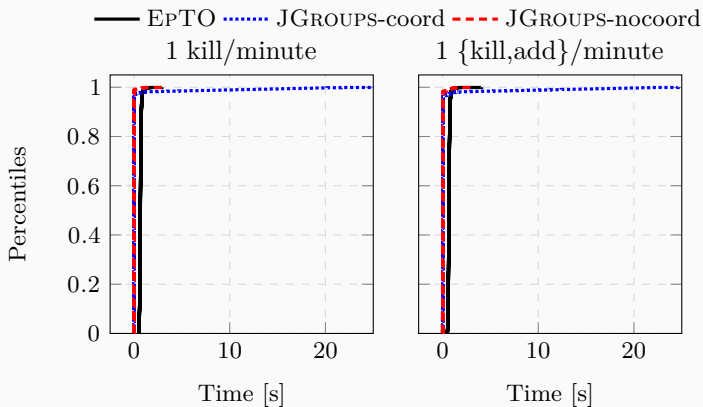
Bandwidth Real Churn



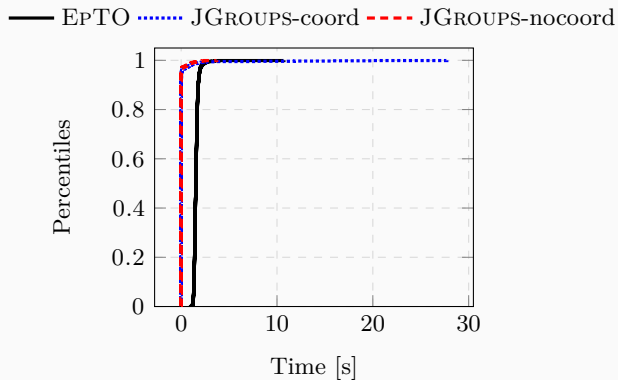
Local Dissemination Stretch



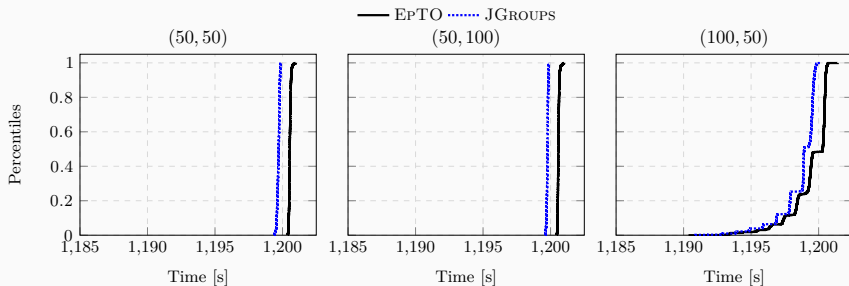
Local Dissemination Stretch Synthetic Churn (100, 50)



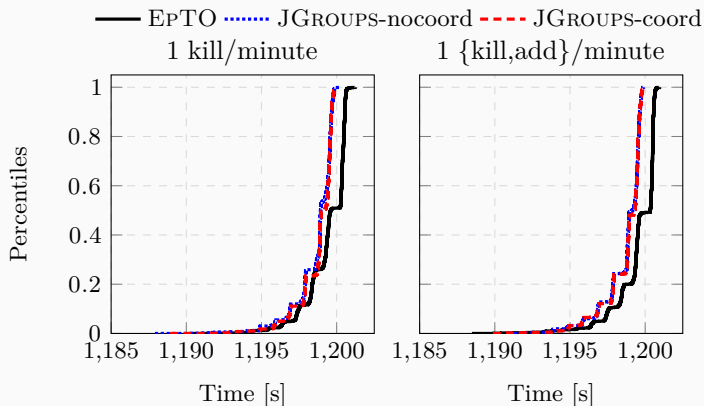
Local Dissemination Stretch Real Churn



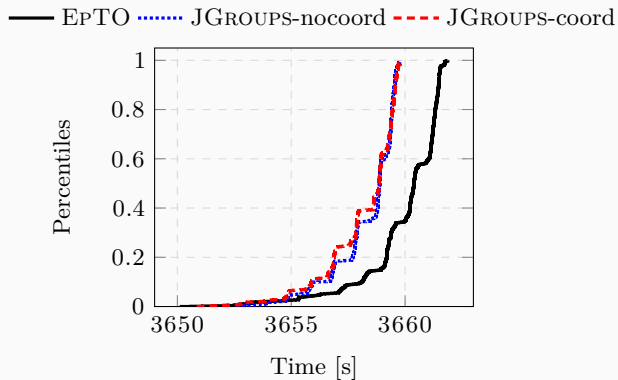
Local Times



Local Times Synthetic Churn (100, 50)



Local Times Real Churn



Protocol		Cluster parameters		
		(50, 50)	(50, 100)	(100, 50)
EPTO	Receive	10.84 ± 0.16	22.31 ± 0.39	26.01 ± 0.27
	Sending	10.84 ± 0.16	22.31 ± 0.39	26.01 ± 0.27
JGROUPS	Receive	0.78 ± 0.03	1.45 ± 0.01	1.88 ± 0.01
	Sending	0.77 ± 0.03	1.44 ± 0.01	1.84 ± 0.01

Total events sent in a stable environment

Protocol	Cluster parameters		
	(50, 50)	(50, 100)	(100, 50)
EPTO	$59\,993.8 \pm 3.3$	$119\,898.2 \pm 9.7$	$59\,913.0 \pm 164.3$
JGROUPS	$59\,961.9 \pm 10.9$	$119\,885.7 \pm 5.0$	$60\,023.1 \pm 287.1$

Total events sent with a synthetic churn (100, 50)

Protocol	Cluster parameters	
	1 kill/minute	1{kill,add}/minute
EPTO	53 898.5 \pm 133.9	59 798.6 \pm 140.1
JGROUPS-coord	53 834.7 \pm 175.5	59 507.9 \pm 240.9
JGROUPS-nocoord	53 830.5 \pm 200.3	59 450.5 \pm 175.1

Total events sent during a real trace

Protocol	Events sent
EPTO	165 844.2 \pm 210.2
JGROUPS-coord	166 183.0 \pm 1368.1
JGROUPS-nocoord	166 585.8 \pm 824.9

Conclusion

- Difference not strong enough at 100 peers scale
- High CPU usage
- Docker problems on AWS/GCE

- Obtain more resources to have stronger results
- Implement a Push-Pull EPTO version
- Use Kubernetes instead of Docker + Docker swarm
- Refine Framework Architecture

Questions?

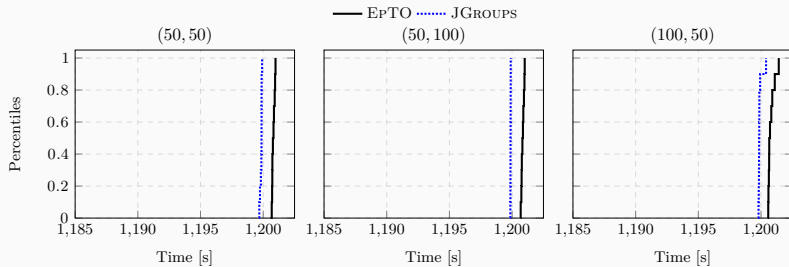
Total GB sent/received Synthetic Churn (100, 50)

Protocol		Churn parameters	
		1 kill/minute	1{kill,add}/minute
EPTO	Receive	21.00 \pm 0.24	26.32 \pm 0.32
	Sending	21.21 \pm 0.25	26.57 \pm 0.32
JGROUPS-coord	Receive	1.47 \pm 0.02	1.75 \pm 0.02
	Sending	1.43 \pm 0.02	1.70 \pm 0.02
JGROUPS-nocoord	Receive	1.45 \pm 0.01	1.73 \pm 0.02
	Sending	1.41 \pm 0.01	1.68 \pm 0.02

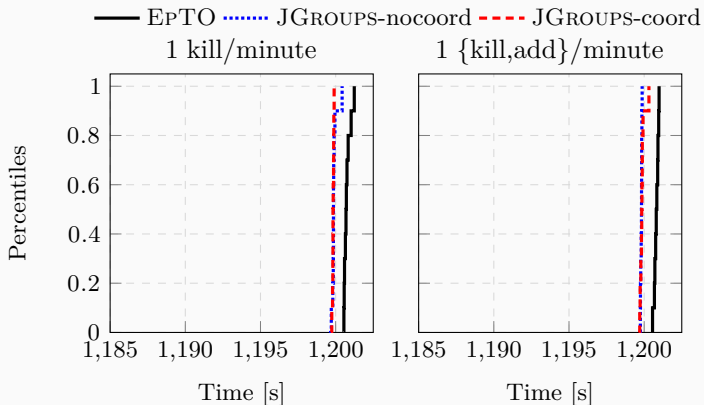
Total GB sent/received Real Churn

Protocol		Churn parameters
		Real Trace
EPTO	Receive	81.41 ± 1.08
	Sending	82.67 ± 1.08
JGROUPS-coord	Receive	5.56 ± 0.08
	Sending	5.40 ± 0.08
JGROUPS-nocoord	Receive	5.58 ± 0.05
	Sending	5.43 ± 0.05

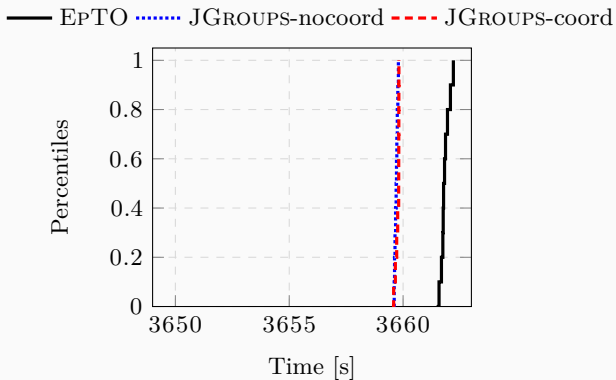
Global Times



Global Times Synthetic Churn



Global Times Real Churn



Dissemination Example

Round 0

