DeSceNt:
Plug-based Decentralized Social Network

AELOS - ASAP – CIDRE- GDD
Pascal Molli - University of Nantes - GDD
May 2018

https://descent.cominlabs.u-bretagneloire.fr/

Overview

DeSceNt aims to ease the writing of distributed programs on a federation of plug computers.

- Plugs: cheap computers (35$)
- Federation: social interconnection of Plugs
- Writing programs for such infrastructure

Why?

Give more control to people on their digital life by providing Decentralized applications:

- privacy preservation,
- resistance to censorship,
- Data owners are free to store their data where they want.

http://abiteboul.com/DOCS/14.pims.pdf

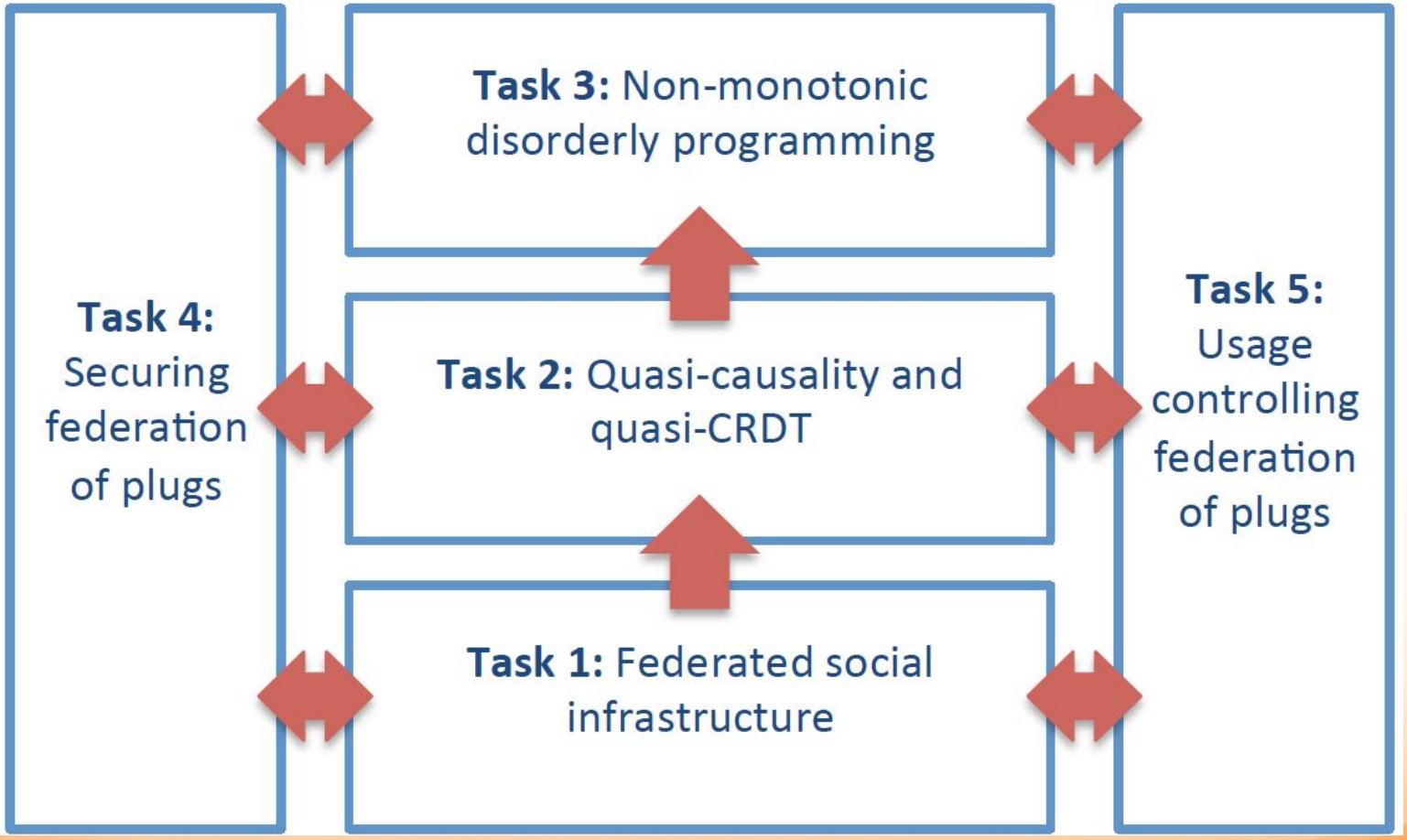
Use cases

- An equivalent of Google drive services (doc, files)
- Dropbox-like services,
- Distributed social networks (Diaspora…),
- Distributed recommendation systems (Whatsup),
- Federated Datastore…

What We Need ?

- Reliable communications between plugs ***(Task1)***:
  - publish-subscribe, multicast etc…
- Federated data structures and consistency criteria to write programs ***(Task2&3)***
  - Plugs communicate through shared data structures edited concurrently. What kind of consistency we can afford?
- We need security enforcement and usage control of this social infrastructure ***(Task 4&5)***
  - How to monitor decentralized infrastructure for attacks?
  - What the users are doing with my data?

DeSceNt Tasks



Writing Decentralized Apps ?

```
import cods.uc;
void main() {
    /* configure the network
    Network.configure(network);
    /* connect to the object
    Register x = new Register();
    Register x = UC.connect!Register("x");
    /* use x normally
    x.write(x.read() + 1);
}
```

Connect a secure federation
Thanks to task1,4 and 5

Access a federated data structure with weak consistency
Criteria (Update Consistency) thanks to task 2&3

D program from task 3

ComInLabs

Use-Case Distributed Collaborative Editor

How to run a GoogleDoc-like application as a full decentralized application ?

Google Doc is great… but…

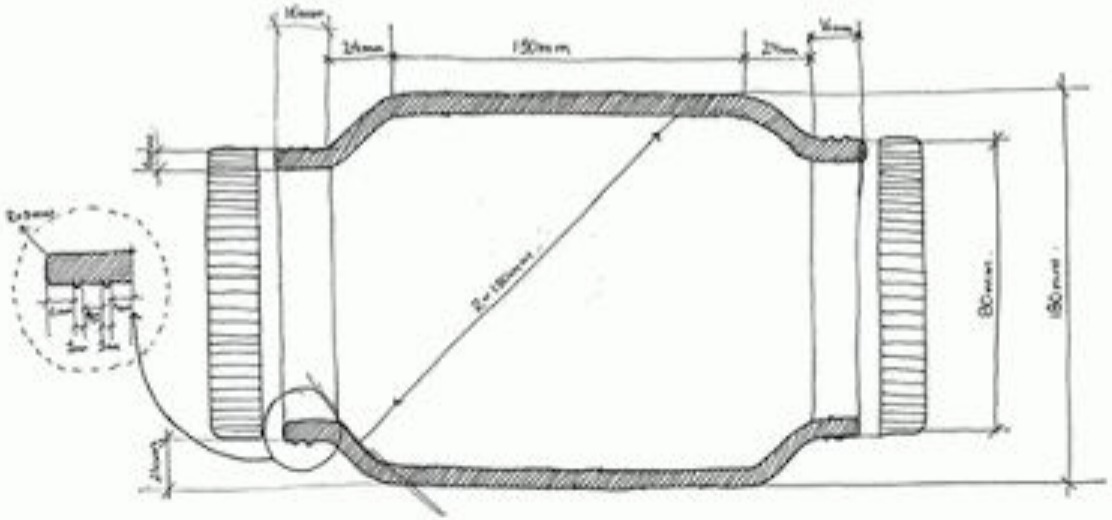
Product Proposal

Double Lid Peanut Butter Jar

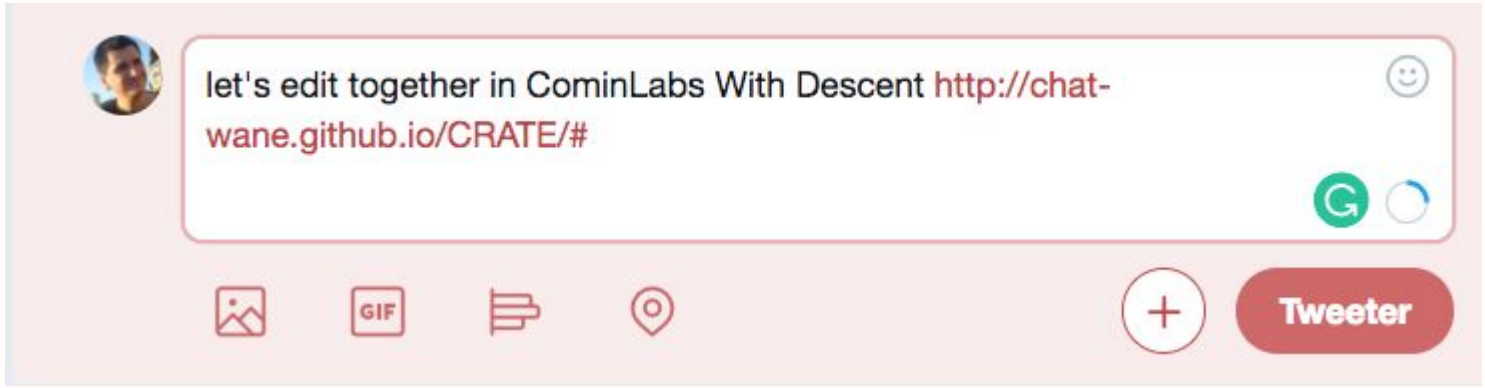
Project status:

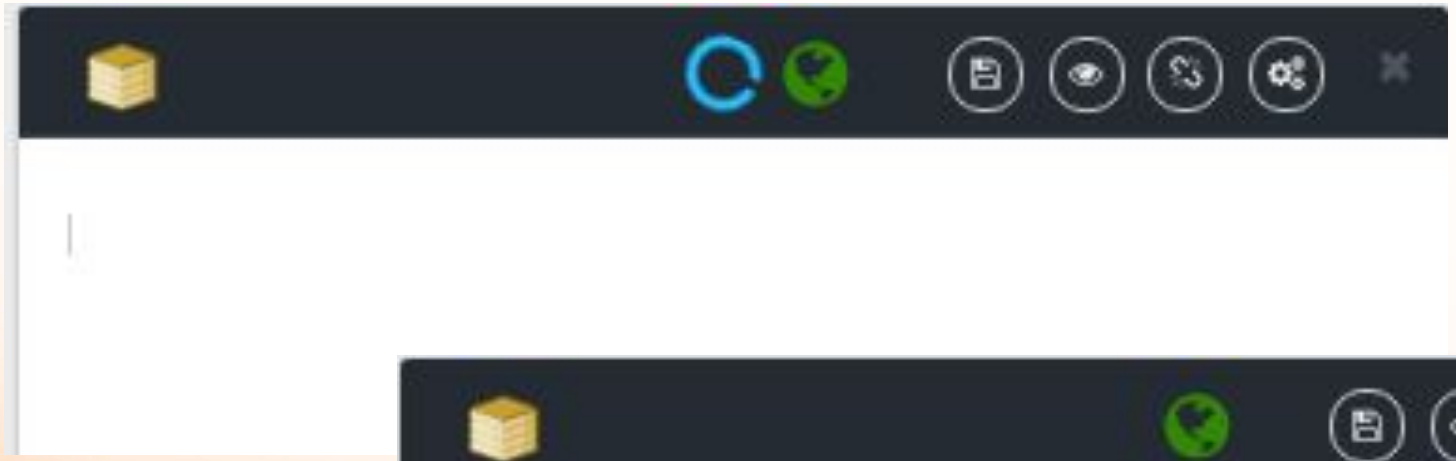
PROBLEM STATEMENT

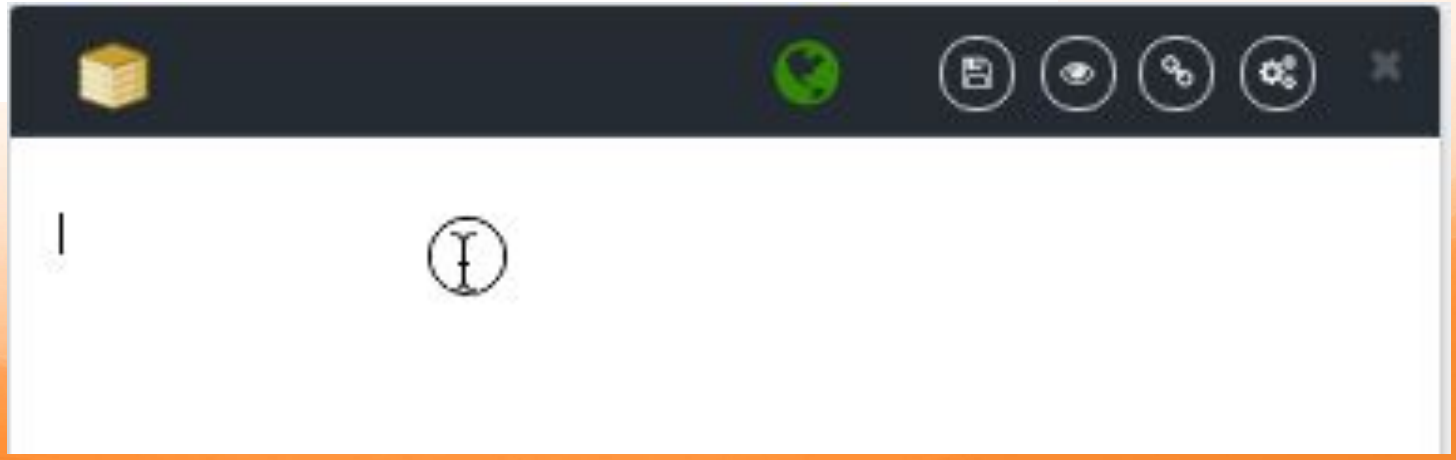
Even for the tiniest hands, it is sometimes difficult to get that last bit of peanut butter out of the bottom of the jar.



We want massive collaborative editing, with no server…
just millions of people with their browsers and plugs.

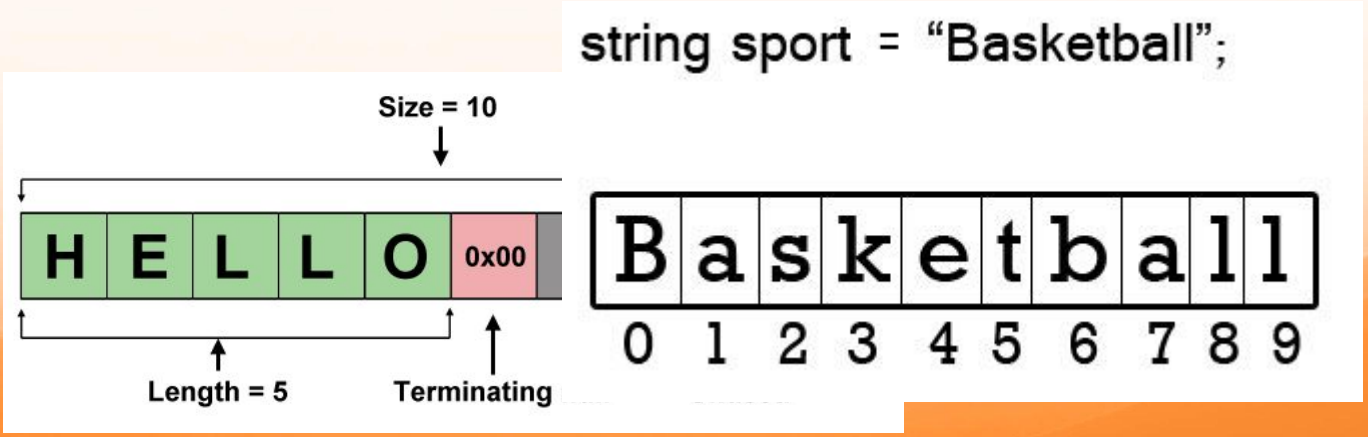

let's edit together in CominLabs With Descent http://chat-wane.github.io/CRATE/#
GIF
Tweeter





Use-case: Distributed Collaborative Editor

- Each site has a copy of the shared sequence
  - Each site freely updates locally its own copy
  - Broadcast to others *(thanks to task1,4,5 ;)*
  - Each site integrates remote operations
  - System is correct if it at least eventually converges *(or more see task3 ;)*



Consistency

What we have to read in the sequence when edited concurrently ?

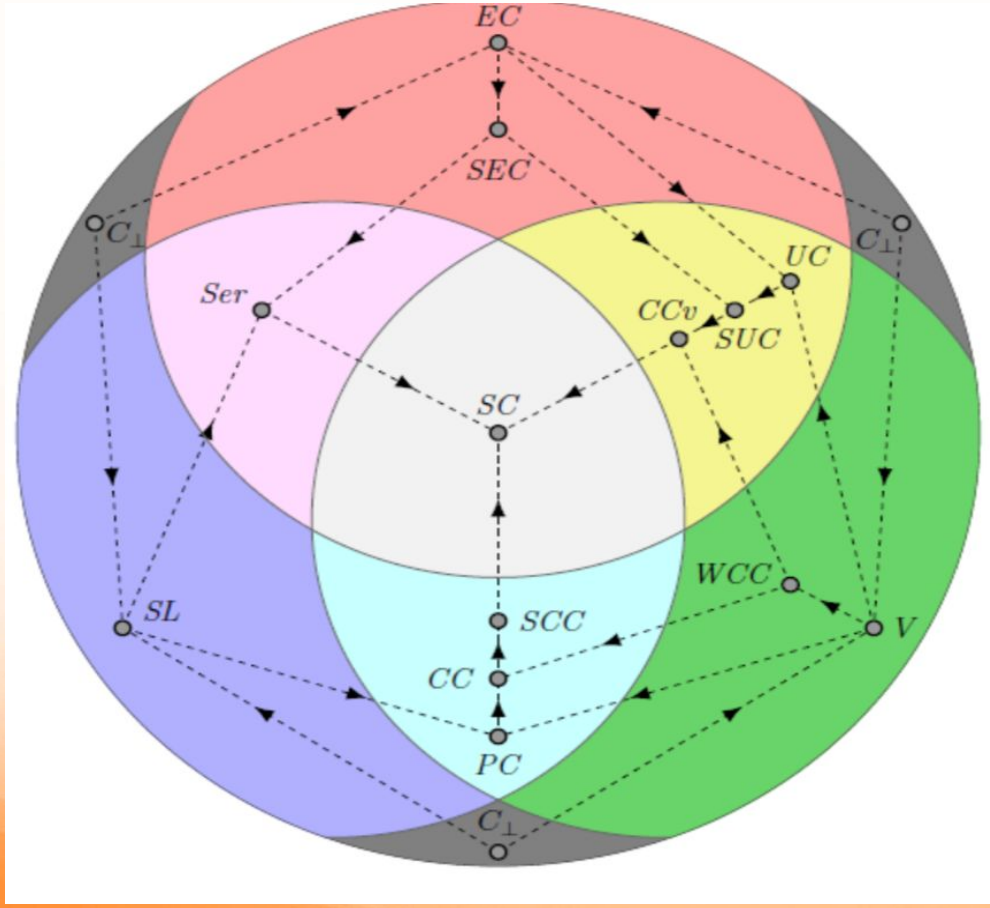
The right shared sequence specification

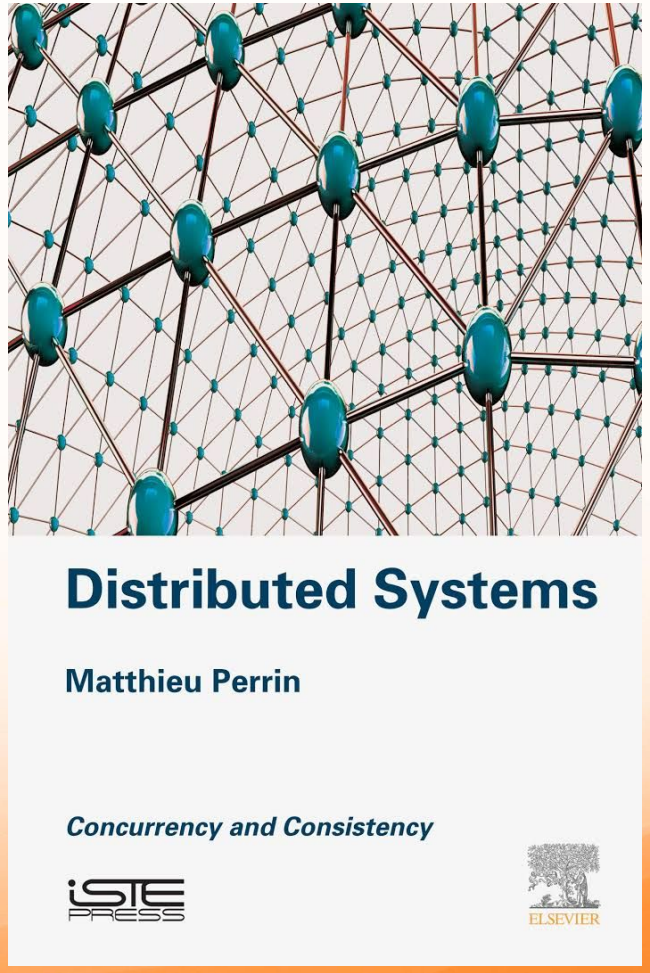
- A sequence is a set of pairs *(id,element)* where *Id* belong to a **dense order**.
  - An order of a set X **is dense** if, for all *x, y* in *X* for which $x < y$, Il existe *z* in *X* st. $x < z < y$
  - Real number is a dense order
- S={(2.5,'a'),(2.6,'b')}
- S.Insert(2.5,'x',2.6) : {(2.5,'a'),(2.55,'x'),(2.6,'b')}
- S.Delete(2.55) : {(2.5,'a'),(2.6,'b')}

Weak Consistency Zoo

- The system is correct if:
  - ~~Intentions, causality, eventual consistency , Strong Eventual consistency, causal consistency, PRAM consistency, Serializability~~
- Update-consistency !!
  - The shared object converges to a state reachable by the sequential specification.
  - Perrin, M., Mostefaoui, A., & Jard, C. (2015, May). Update consistency for wait-free concurrent objects. In Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International (pp. 219-228). IEEE

Breakthrough : Formalization of weak consistency world !





Distributed Collaborative Editor

- Run in all browsers:

```
Import cods.ec
void main() {
  Network.configure(network)
  Seq l = UC.connect!Seq("l");
  l.insert('x',1);
}
```

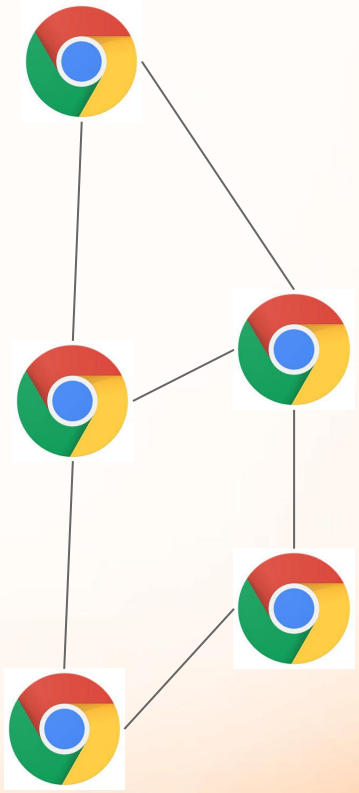
Communication

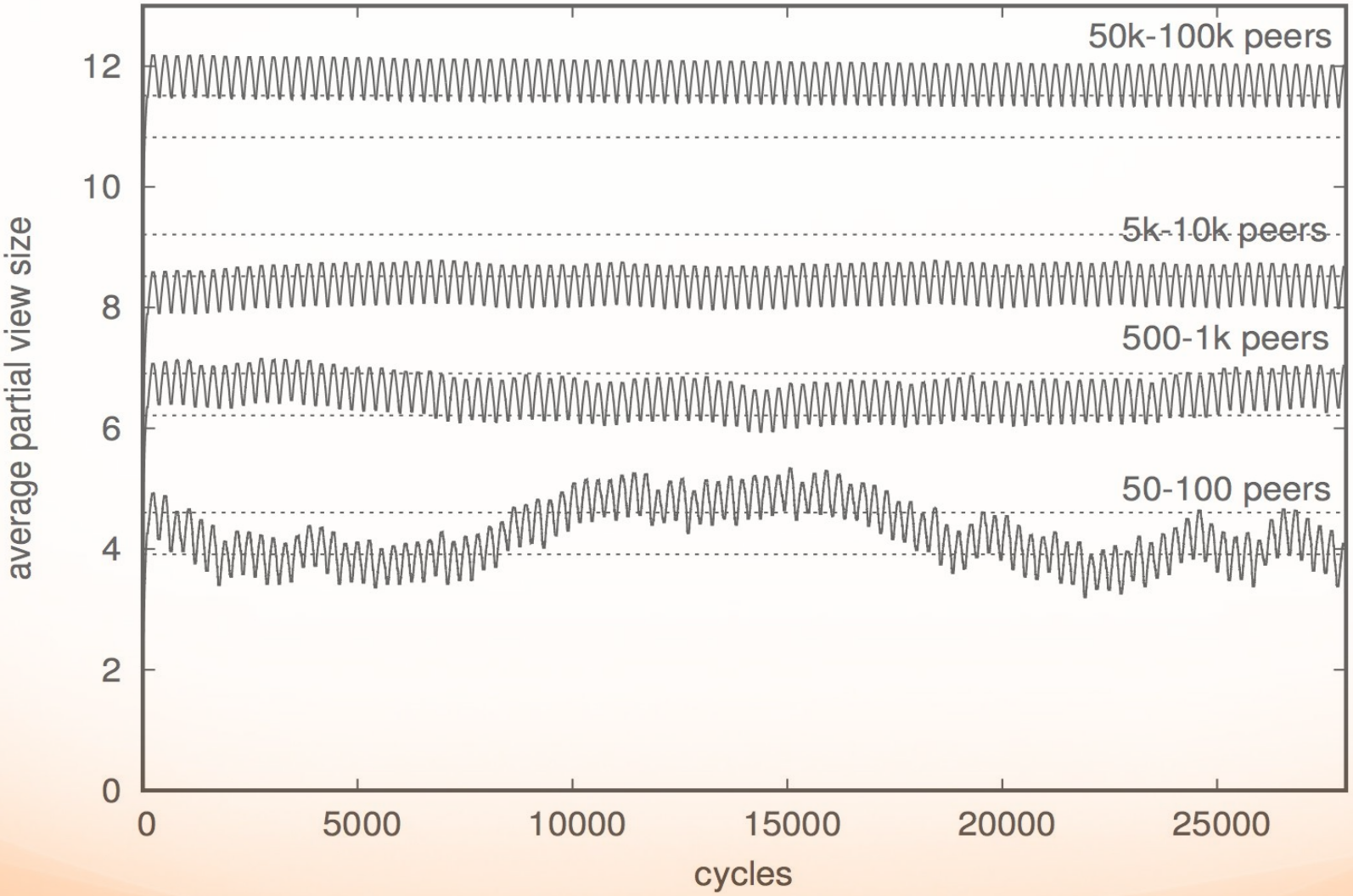
How to connect browsers with reliable and causal broadcast?

Gossiping on a network of browsers

- Adaptive gossiping on top of WebRTC:
  - Nédelec, B., Tanke, J., Frey, D., Molli, P., & Mostéfaoui, A. (2017). An adaptive peer-sampling protocol for building networks of browsers. *World Wide Web Journal*, 1-33.
  - SPRAY:

    https://github.com/RAN3D/spray-wrtc
- **Ensure to broadcast to log(n) participants, where n is the approximate number of people at a given time.**





(b) Average size of partial views over long period of time.

Breakthrough:
Breaking the causality barrier

- Need that delete(x) arrive after insert(x), but cannot afford vector clocks
- Full causality without communicating and storing vector clocks.
  - Brice Nédelec, Pascal Molli and Achour Mostéfaoui. Breaking the Scalability Barrier of Causal Broadcast for Large and Dynamic Systems - Submitted to SRDS2018.
  - Brice Nédelec, Pascal Molli, Achour Mostefaoui. The Last Monotonic Bound of Causal Broadcast - Submitted to DISC2018.

Breaking the Scalability Barrier of Causal Broadcast for Large and Dynamic Systems

TABLE I: Space and time complexity of causal broadcast protocols. N is the number of processes. W is the number of received messages awaiting for delivery. P is the number of messages that are still unsafe to be purged. B is the size of a set of temporary buffers.

	dynamic	message overhead	local space consumption	delivery execution time
vector-based [7]	✓	$O(N)$	$O(N+W.N)$	$O(W.N)$
FIFO+forward [14]	✗	$O(1)$	$O(P+W)$	$O(1)$
this paper	✓	$\mathbf{O(1)}$	$\mathbf{O(N+B+W)}$	$\mathbf{O(1)}$

Distributed Collaborative Editor

- Run in all browsers:

```
Import cods.ec
void main() {
  Network.configure(Causal+SPRAY)
  Seq l = UC.connect!Seq(“l”);
  l.insert(‘x’,1);
}
```

Shared Data Structure

How to implement efficiently shared data structures on decentralized applications?

Does the shared editor scale ??

- Each site has a copy of the shared sequence
  - Each site freely updates locally its own copy
    - **Time complexity to generate operation ?**
    - **Space complexity of messages and local state ?**
  - Broadcast to others *(thanks to task1,4,5 ;)*
    - **Communication complexity? |message| * #participants**
  - Each site integrates remote operations
    - **Time complexity to integrate operation**
- **System scales if all complexities are sublinear to the number of participants and/or to the size of documents**

Breakthrough: Implementing the dense order...

- Implementing a dense order: assign a unique ID to every character. The size of the ID can be bounded to $O(\log(n)^2)$
  - Nédelec, B., Molli, P., & Mostéfaoui, A. A scalable sequence encoding for collaborative editing. Concurrency and Computation: Practice and Experience.

- The CRATE editor : https://github.com/Chat-Wane/CRATE
  - Nédelec, B., Molli, P., & Mostefaoui, A. (2016, April). Crate: writing stories together with our browsers. In Proceedings of the 25th International Conference Companion on World Wide Web (pp. 231-234). International World Wide Web Conferences Steering Committee.

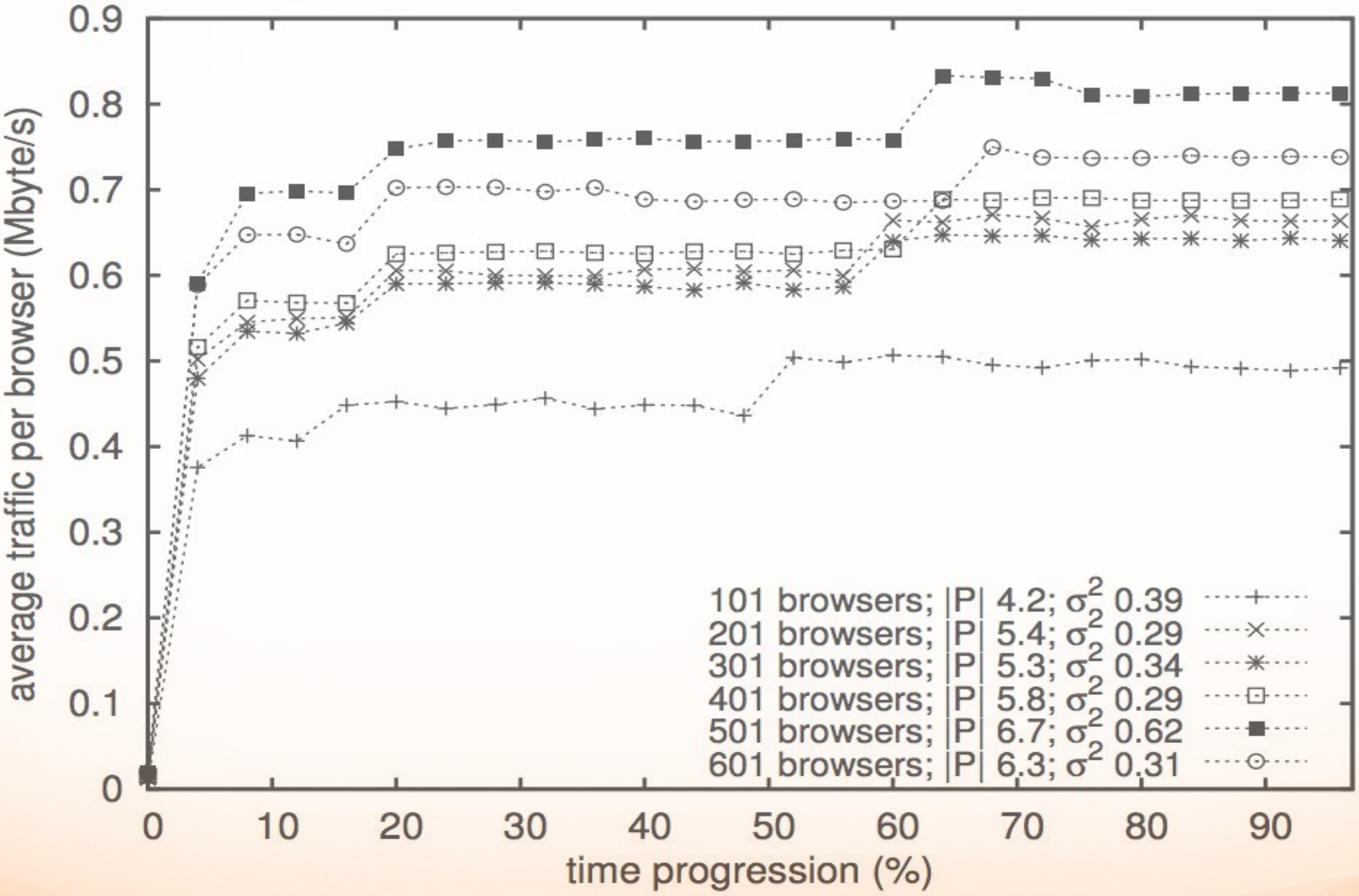


Figure 4: Average traffic per second.

Distributed Collaborative Editor

- Run in all browsers:

```
Import cods.ec
void main() {
  Network.configure(Causal+SPRAY)
  LSeq l = UC.connect!LSeq("l");
  l.insert('x',1);
}
```

Security

How to secure decentralized applications?

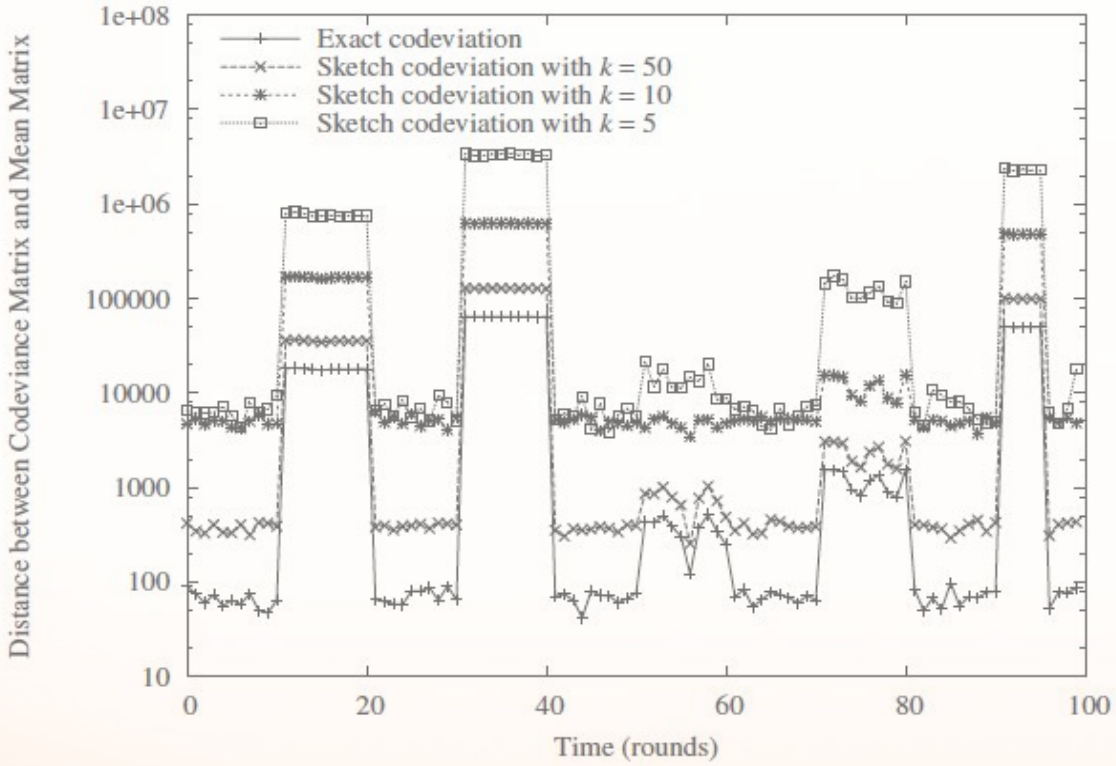
T4: Selected Results

- Emmanuelle Anceaume, Yann Busnel, Deviation Estimation between Distributed Data Streams. In the 10th European Dependable Computing Conference (EDCC 2014), Newcastle upon Tyne, UK, May 2014.
- Emmanuelle Anceaume, Yann Busnel. Lightweight Metric Computation for Distributed Massive Data Streams. Dans Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS), LNCS, Springer, 10430(33):1-39, 2017.

Distributed Monitoring Application

- Co-variance analysis of incoming streams is a powerful approach to detect malicious attacks whatever the pattern of the attack.
- In decentralized setting, is it possible to compute equivalent metrics with limited resources ?
- **Yes ! Co-deviation computation with limited resources**

Detection of Different Profiles of Attacks



- Distance $\|\Sigma_r - \mathbb{E}(\Sigma_r)\|$ as a function of round r with $\mathbb{E}(\Sigma_r) = ((r-1)\mathbb{E}(\Sigma_{r-1}) + \Sigma_r)/r$
- Different scenario of attacks have been conducted on $s = 10$ sites.

Conclusions & Perspectives

- DeScENt achieved important results:
  - Propose a way to write and execute decentralized applications in a network of web browsers
  - Breakthrough in CRDT, Causality management, consistency formalization.
  - Investigate original way in security management and usage control.
- **Demonstrated it is possible to build some decentralized applications that outperform centralized ones.**

Descent Followups

- **WeBrowse :** Web of Browsers 2017-2018 (ASAP/APIZEE)
  - Build an ephemeral web with no servers.
- **CIFRE OpenDataSoft** (2017-2020)
  - Usage Control Policies on Open Data Platform.
- **Project ANR O'Browser:**
  - Opportunistic Fog Computing with Browsers (2017-2020)
  - *"Imagine designing and deploying a distributed application on millions of machines simply by posting a link on Twitter or by buying a word on Google Adwords."*

Thanks CominLabs !

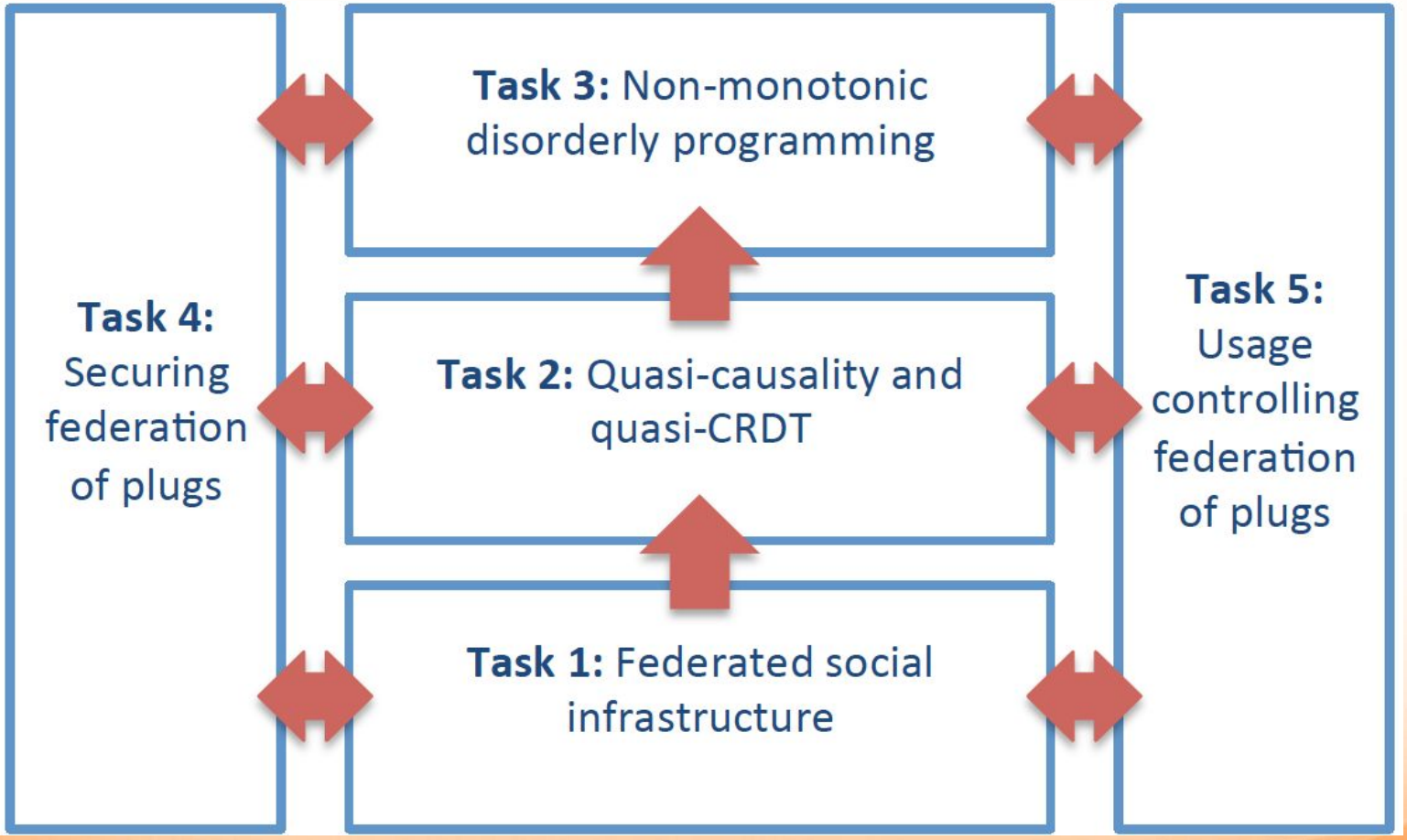
- CominLabs allowed us to take risks !
- We submitted a project where we are not sure to succeed
- Big Risk, Big Gains ;)

Questions

Task 3: Non-monotonic disorderly programming (2015)

- PERRIN, Matthieu, MOSTÉFAOUI, Achour, et JARD, Claude. Brief Announcement: Update Consistency in Partitionable Systems. DISC2014, p. 546.

DeSceNt Tasks



Task 2: Quasi causality and CRDT (2015)

- LSEQ for distributed collaborative editing
  - Brice Nédelec, Pascal Molli, Achour Mostefaoui, Emmanuels Desmontils, LSEQ: an adaptive structure for sequences in distributed collaborative editing, Proceeding DocEng '13 Proceedings of the 2013 ACM symposium on Document engineering Pages 37-46, 2013 (Best Student Paper)

Task 3: Non-monotonic disorderly programming

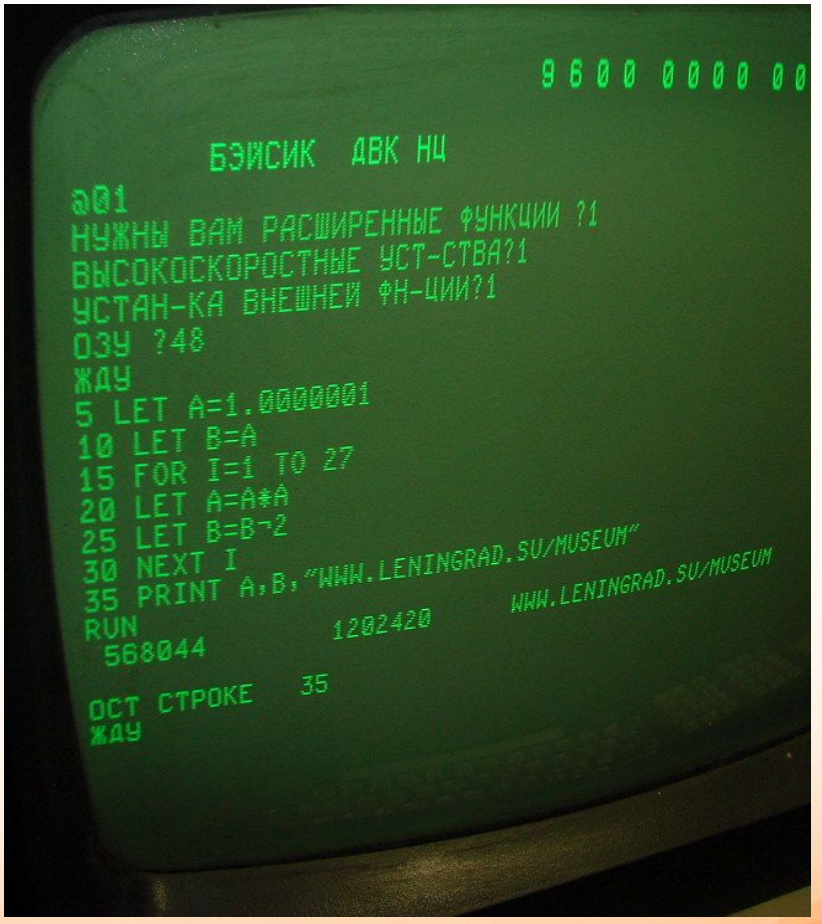
- Perrin, M., Mostefaoui, A., & Jard, C. (2015, May). Update consistency for wait-free concurrent objects. In Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International (pp. 219-228). IEEE.
- Matthieu Perrin, Achour Mostéfaoui, Claude Jard: Causal consistency: beyond memory. PPOPP 2016: 26:1-26:12
- Matthieu Perrin, Matoula Petrolia, Achour Mostéfaoui, Claude Jard, On Composition and Implementation of Sequential Consistency. DISC 2016: 284-297
- Davide Frey, Achour Mostéfaoui, Matthieu Perrin, Pierre-Louis Roman, François Taïani - Speed for the Elite, Consistency for the Masses: Differentiating Eventual Consistency in Large-Scale Distributed Systems. SRDS

Principles of Sequence CRDT

- Encode the order of the sequence in the Id of elements (remember ;)

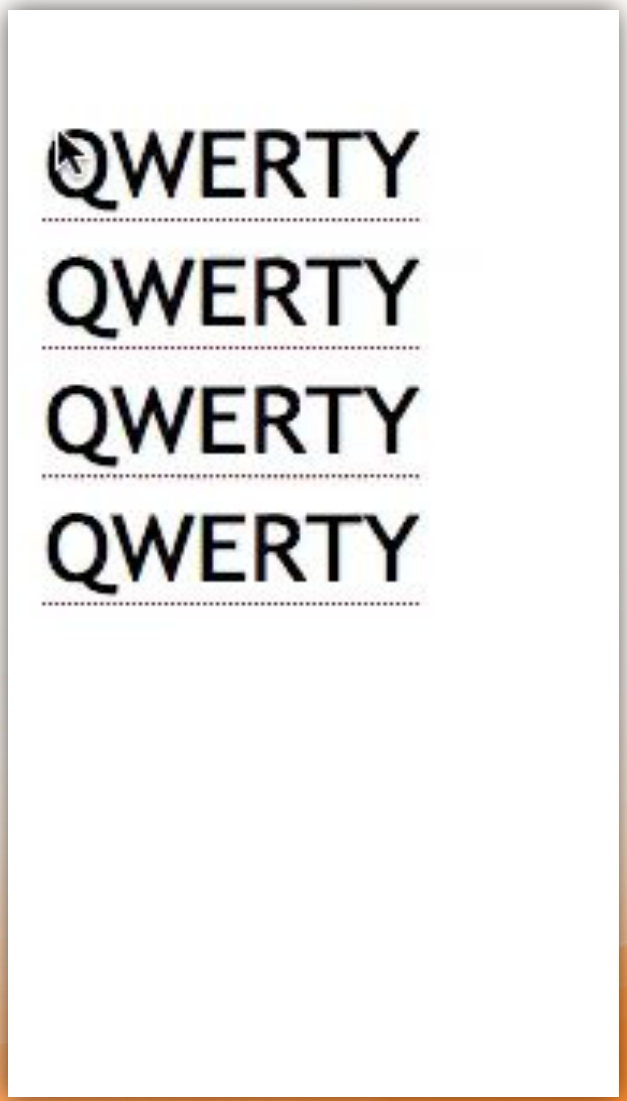
```
10 LET B=A
15 For I=1 to 27
20 LET A=A*A
21 NEXT I
```

- Arghh, I forgot *LET B=B^2* before *NEXT I*, no way to use 20,5 ??



Scientific problem

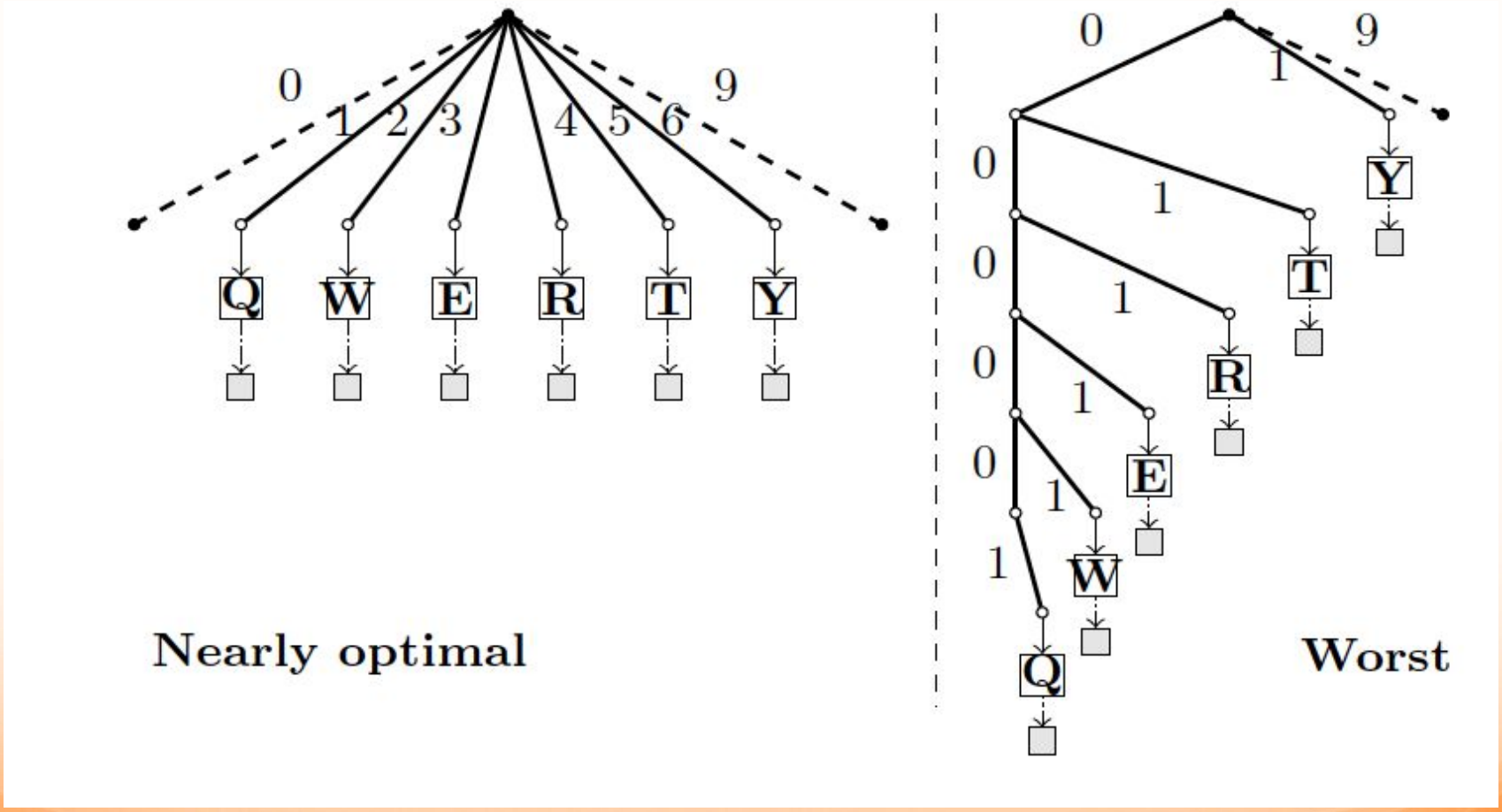
- Write an allocation strategy ID for sequence element that is independent of insertion order
- Many ways to type “QWERTY”, how to compute the smallest IDs for each character whatever insertion order ?



Algorithm 1 General outlines of variable-size identifiers CRDTs for sequences

1: **INITIALLY:**
2: $\mathcal{T} \leftarrow \varnothing$; ▷ structure of the CRDT for sequences
3:
4: **LOCAL UPDATE:**
5: **on** insert $(p \in \mathcal{I}, \alpha \in \mathcal{A}, q \in \mathcal{I})$:
6: **let** $path \leftarrow allocPath(p.P, q.P)$;
7: **let** $dis \leftarrow allocDis(p, path, q)$;
8: $broadcast('insert', \langle path, \alpha, dis \rangle)$;
9: **on** delete $(i \in \mathcal{I})$:
10: $broadcast('delete', i)$;
11:
12: **RECEIVED UPDATE:**
13: **on** insert $(i \in \mathcal{I})$: ▷ once per distinct triple in $\mathcal{I}$
14: $\mathcal{T} \leftarrow \mathcal{T} \cup i$;
15: **on** delete $(i \in \mathcal{I})$: ▷ after the remote $insert(i)$ is done
16: $\mathcal{T} \leftarrow \mathcal{T} \setminus i$;
17:

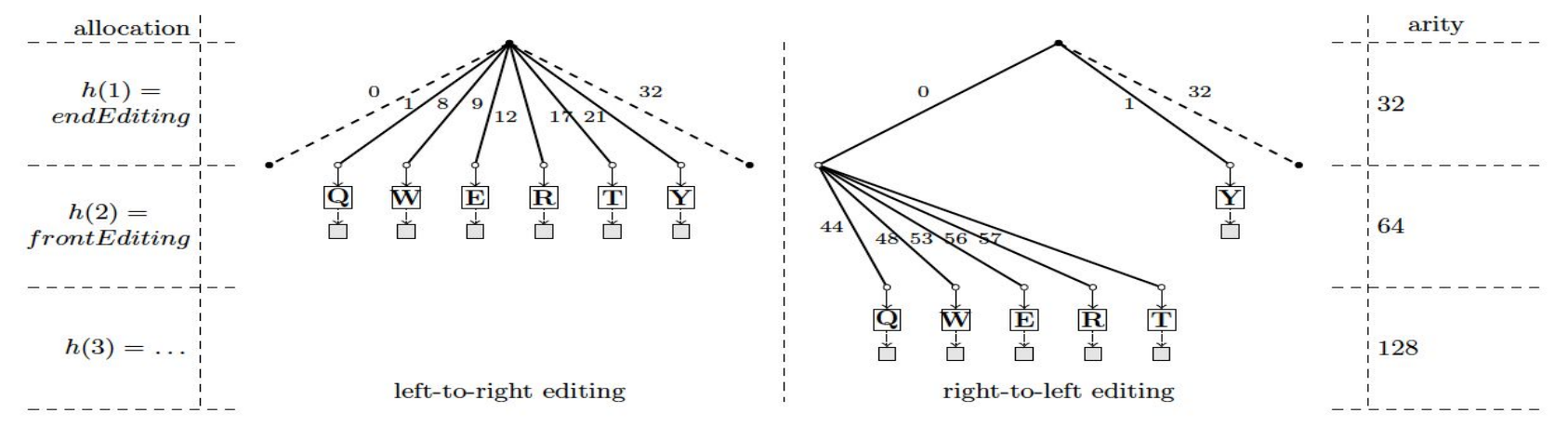
PB: Order of Insertions



Typed: Q;W;E;R;T;Y

Typed: Y;T;R;E;W;Q

Combine Exponential tree & random allocation



$O((\log n)^2)$ -> avoid to rebalance IDs…

Evaluation

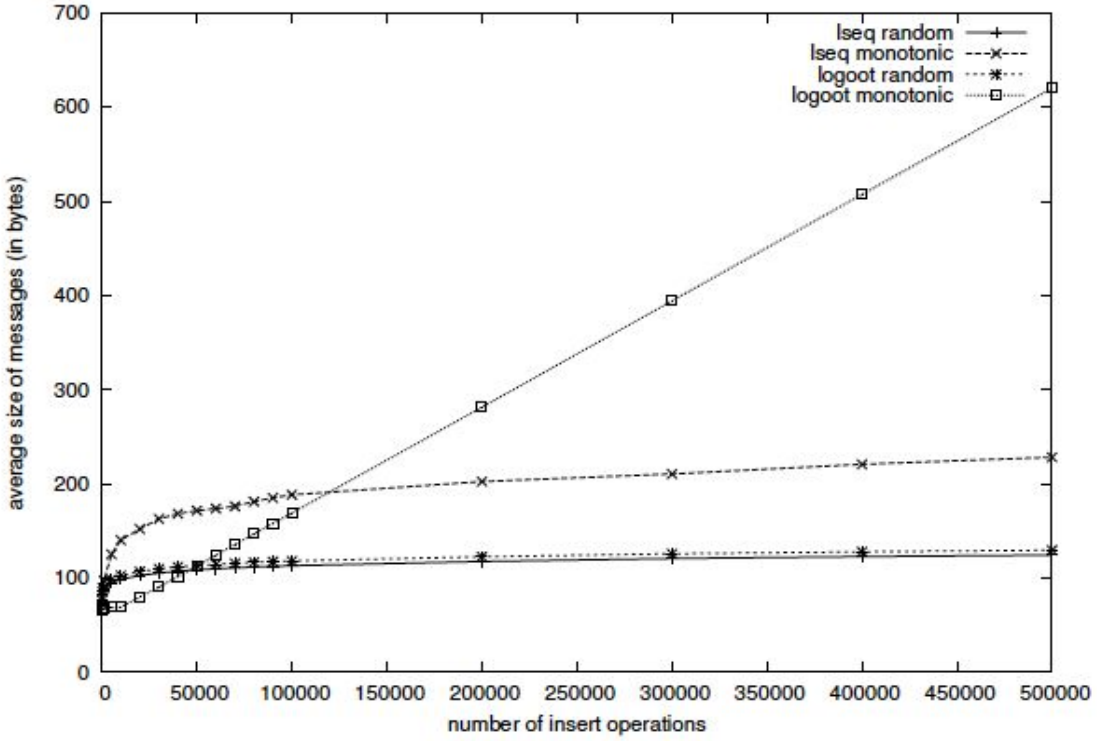


Figure 6: Average size of the messages exchanged between two peers during repeated insertions of elements in the sequence. The measurements concern two editing behaviours (monotonic and random) with two CRATE using different CRDTs for sequences (Logoot and LSEQ-based). The replicated document grows up to half a million characters.

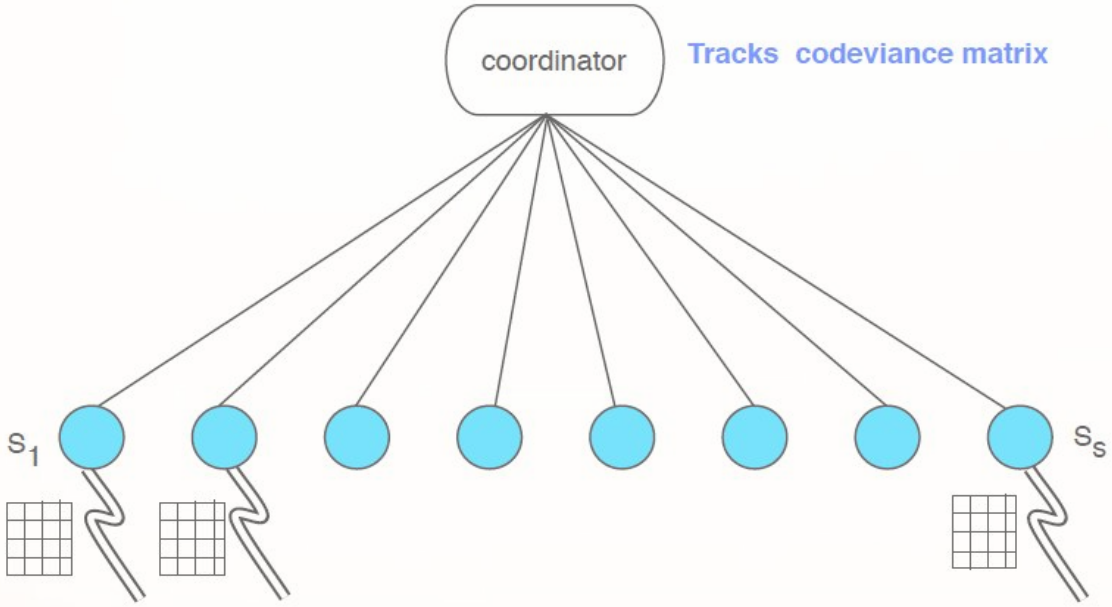
Using LSEQ...

```
Import cods.ec
void main() {
  Network.configureCausal(network)
  LSeq l = EC.connect!Lseq(“l”);
  l.insert(‘x’,1);
}
```

Task 2: Selected Results

- Nédelec, B., Molli, P., & Mostéfaoui, A. A scalable sequence encoding for collaborative editing. Concurrency and Computation: Practice and Experience.
- Nédelec, B., Molli, P., & Mostefaoui, A. (2016, April). Crate: writing stories together with our browsers. In Proceedings of the 25th International Conference Companion on World Wide Web (pp. 231-234). International World Wide Web Conferences Steering Committee.
- Mostéfaoui, A., & Weiss, S. (2017, September). Probabilistic Causal Message Ordering. In International Conference on Parallel Computing Technologies (pp. 315-326). Springer, Cham.
- Brice Nédelec, Pascal Molli and Achour Mostéfaoui. Breaking the Scalability Barrier of Causal Broadcast for Large and Dynamic Systems - Submitted to SRDS2018.

Computing Codeviance Between Streams



- Goal of the coordinator: computing the codeviance matrix, which represents the dispersion matrix of the s streams.
- Let $\mathbb{X} = \{X_1, X_2, \ldots, X_s\}$ be the set of fingerprint vectors $X_1, \ldots, X_n$ describing respectively the streams $\sigma_1, \ldots, \sigma_s$.

$$\widehat{\Sigma} = \left[\widehat{\mathrm{cod}}(X_i, X_j)\right]_{1 \leq i \leq s, 1 \leq j \leq s}$$

Computing Codeviance…

- The algorithm proceeds in rounds
- In round $r = 1$, each site computes a sketch based on the first d_1 items of its stream and sends it to the coordinator
- Upon receipt of the first sketch, the coordinator asks for the $s - 1$ other sketches and builds the codeviance matrix $\widehat{\Sigma} = \left[\widehat{\text{cod}}_1(X_i, X_j)\right]_{1 \leq i \leq s, 1 \leq j \leq s}$
- In round $r' = r + 1$, each site computes a sketch based on the subsequent $d' = 2d_r$ items of its stream and sends it to the coordinator
- Upon receipt of the first $(r + 1) - th$ sketch, the coordinator asks for the $s - 1$ other sketches and updatess the codeviance matrix as $\widehat{\Sigma} = \left[\widehat{\text{cod}}_r(X_i, X_j) + \widehat{\text{cod}}_{r+1}(X_i, X_j)\right]_{1 \leq i \leq s, 1 \leq j \leq s}$
- ...

Task4 Selected Results

- Emmanuelle Anceaume, Yann Busnel. Lightweight Metric Computation for Distributed Massive Data Streams. Dans Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS), LNCS, Springer, 10430(33):1-39, 2017.
- Nicolò Rivetti, Yann Busnel, Avigdor Gal. FlinkMan: Anomaly Detection in Manufacturing Equipment with Apache Flink: Grand Challenge. Dans 11th ACM International Conference on Distributed Event-Based Systems (DEBS 2017), Barcelone, Espagne, Juin 2017.
- Nicolò Rivetti, Emmanuelle Anceaume, Yann Busnel, Leonardo Querzoni, Bruno Sericola. Online Scheduling for Shuffle Grouping in Distributed Stream Processing Systems. Dans 17th ACM/IFIP/USENIX 13th International Conference on Middleware (Middleware 2016), Trento, Italie, Décembre 2016.
- Nicolò Rivetti, Yann Busnel, Leonardo Querzoni. Load-Aware Shedding in Stream Processing Systems. Dans 10th ACM International Conference on Distributed Event-Based Systems (DEBS 2016), Ivine, CA, USA, Juin 2016.
- Yves Mocquard, Bruno Sericola, and Emmanuelle Anceaume, "Probabilistic Analysis of Counting Protocols in Large-scale Asynchronous and Anonymous Systems",Proceedings of the 15th IEEE International Symposium on Network Computing and Applications (NCA), 2017.
- Yves Mocquard, Samantha Robert, Bruno Sericola and Emmanuelle Anceaume,

T4&5: Selected Results

- Emmanuelle Anceaume, Yann Busnel. Lightweight Metric Computation for Distributed Massive Data Streams. Dans Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS), LNCS, Springer, 10430(33):1-39, 2017.
- Nicolò Rivetti, Yann Busnel, Avigdor Gal. FlinkMan: Anomaly Detection in Manufacturing Equipment with Apache Flink: Grand Challenge. Dans 11th ACM International Conference on Distributed Event-Based Systems (DEBS 2017), Barcelone, Espagne, Juin 2017.
- Yves Mocquard, Bruno Sericola, and Emmanuelle Anceaume, "Probabilistic Analysis of Counting Protocols in Large-scale Asynchronous and Anonymous Systems",Proceedings of the 15th IEEE International Symposium on Network Computing and Applications (NCA), 2017.
- Yves Mocquard, Samantha Robert, Bruno Sericola and Emmanuelle Anceaume, "Analysis of the Propagation Time of a Rumor in Large-scale Distributed Systems", Proceedings of the 15th IEEE International Symposium on Network Computing and Applications (NCA) 2016. Best student paper
- Desmontils, E., Serrano-Alvarado, P., & Molli, P. (2017, October). SWEEP: a Streaming Web Service to Deduce Basic Graph Patterns from Triple Pattern Fragments. In 16th International Semantic Web Conference.

Descent Follow Ups

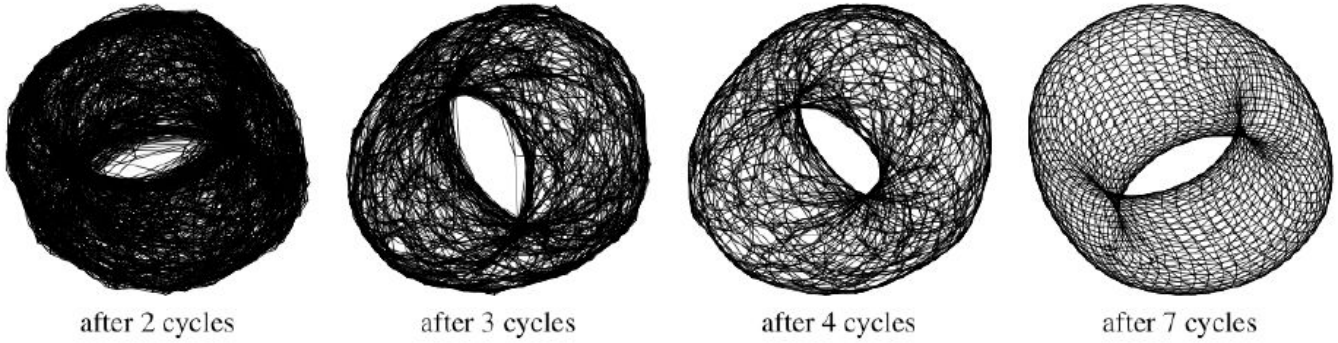
- Project ANR O'Browser: Opportunistic Fog Computing with Browsers (2017-2020)
  - *"Imagine designing and deploying a distributed application on millions of machines simply by posting a link on Twitter or by buying a word on Google Adwords. Imagine doing this without relying on a cloud or a central authority, just through a decentralized execution environment composed of users' browsers that autonomously manages issues such as communication, naming, heterogeneity, and scalability."*

T3: Selected Results

- Perrin, M., Mostefaoui, A., & Jard, C. (2015, May). Update consistency for wait-free concurrent objects. In Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International (pp. 219-228). IEEE.
- Matthieu Perrin, Achour Mostéfaoui, Claude Jard: Causal consistency: beyond memory. PPOPP 2016: 26:1-26:12
- Matthieu Perrin, Matoula Petrolia, Achour Mostéfaoui, Claude Jard, On Composition and Implementation of Sequential Consistency. DISC 2016: 284-297
- Davide Frey, Achour Mostéfaoui, Matthieu Perrin, Pierre-Louis Roman, François Taïani - Speed for the Elite, Consistency for the Masses: Differentiating Eventual Consistency in Large-Scale Distributed Systems. SRDS 2016: 197-206

The polystyrene approach

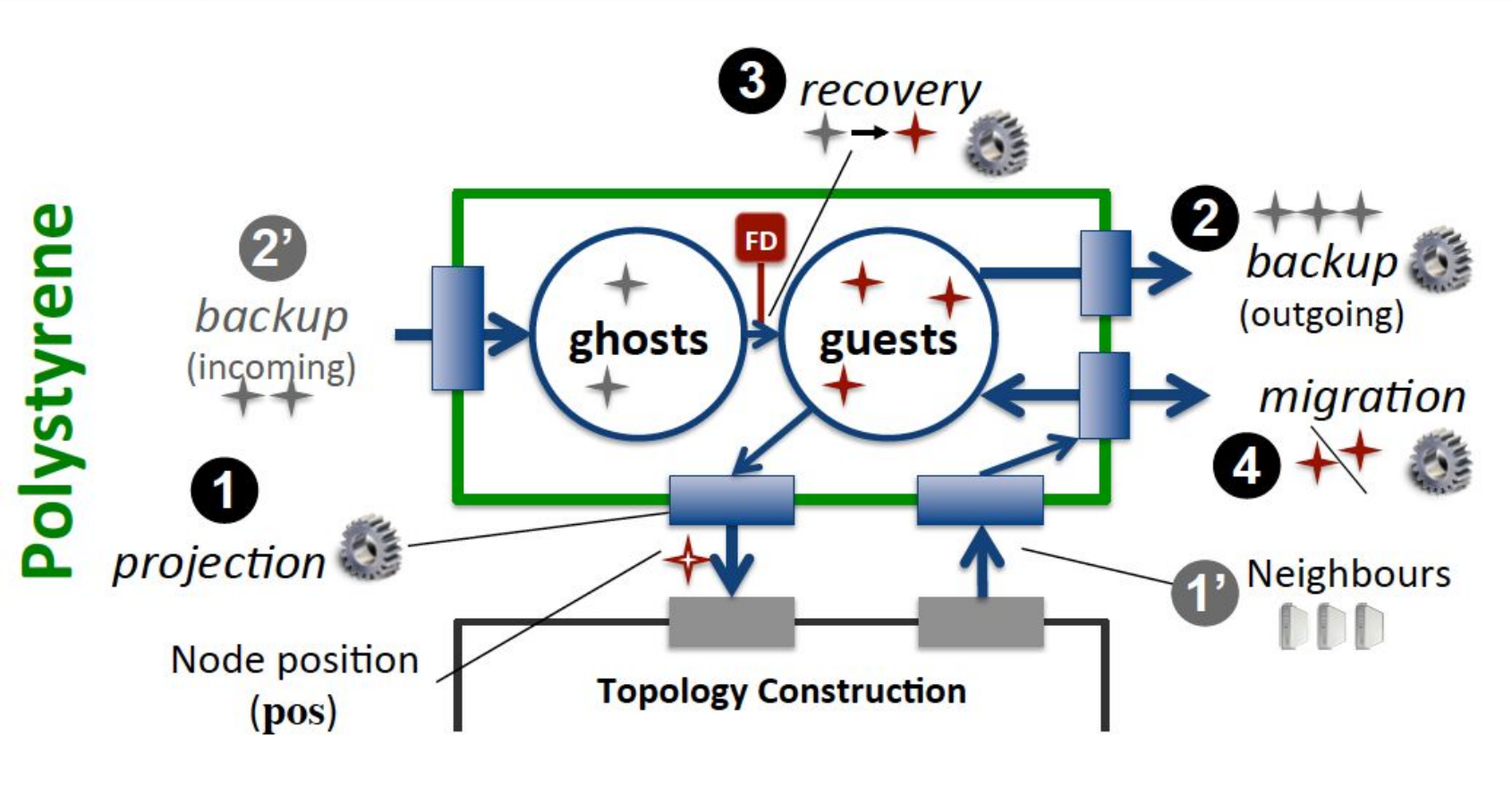
- Epidemic Topology Construction algorithms



Taken from [JMB09]

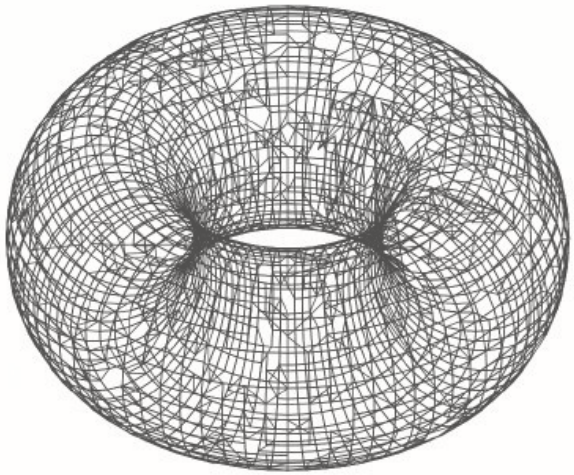
- → Decentralized, fast, scalable
- → Fundamental building block to higher-level services (DHT, Multicast, Pub-Sub, Recommendations)

Polystyrene Approach

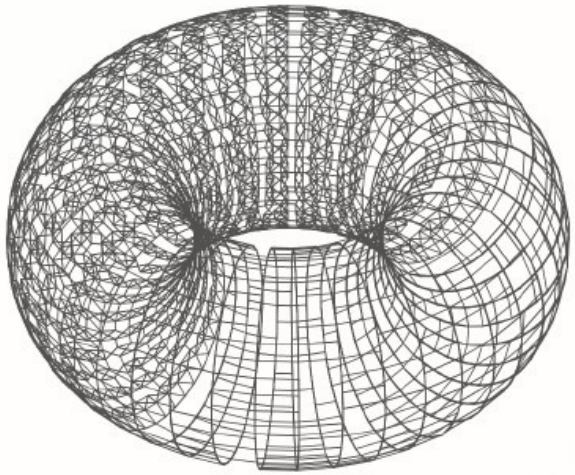


- Ghost -> backup from others, if in failure -> reactivate as guest and re-backup…

Node Reinjection



Polystyrene, r = 125



T-Man, r = 125

Polystyrene recreates uniform shape after repair

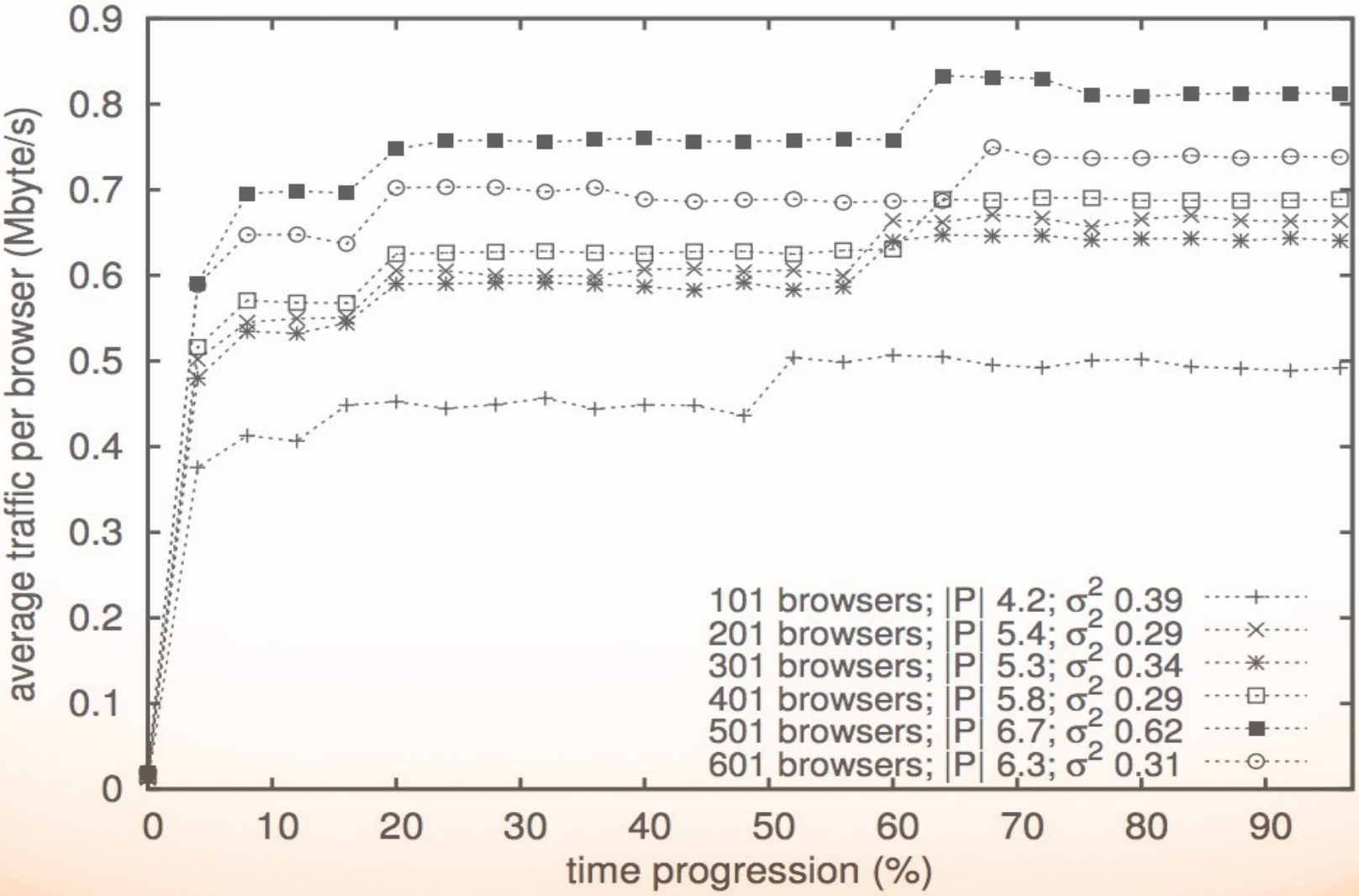


Figure 4: Average traffic per second.

Task2: Quasi-causality and quasi-CRDT.

“Given a federated social infrastructure produced by task 1, the objective is to provide probabilistic causal delivery and probabilistic Conflict-free Replicated Data Types (CRDT) structure such as sequences, sets and graphs.”

Conflict-free Replicated Data Types

- Type where operations cannot *conflict* with any concurrent operation ie.all operations commute:
  - a Set with just insert(e) is a CRDT, but not very useful ;)
- CRDT sequence :
  - Insert(x,1)/insert(y,1) not commute:
    - assign unique ID to element : insert(12<(x->13)<14)
  - ins(<13>)/del(<13>) does not commute:
    - ensure causality -> not happen
- **Challenge: keep ID small, reduce cost of causality**

Related Phd Thesis

- **Nicolo Rivetti**, Started in November 2013. Advisor: Achour Mostefaoui, Co-Advisor: Yann Busnel "Efficient Stream Analysis and its Application to Big Data Processing", defended: 30 septembre 2016
  - Postoc in Israël
- **Brice Nédelec**, Started in September 2012, Advisor: Pascal Molli, Co-Advisor: Achour Mostefaoui "Collaborative editing in browsers", defended 5 october 2016
  - Postdoc in Nantes
- **Matthieu Perrin,** Started in October 2012. Advisor: Claude Jard (AELOS), Co-Advisor: Achour Mostefaoui. "Spécification des objets partagés dans les systèmes répartis sans-attente", defended 7 june 2016
  - MDC Université de Nantes
- Resmi ARIYATTU, Towards federated social infrastructures for plug-based decentralized social networks
- George Nassopoulos, Deducing Basic Graph Patterns from Logs of Linked Data Providers