Trace Synchronization of multi-level, multi-core distributed systems

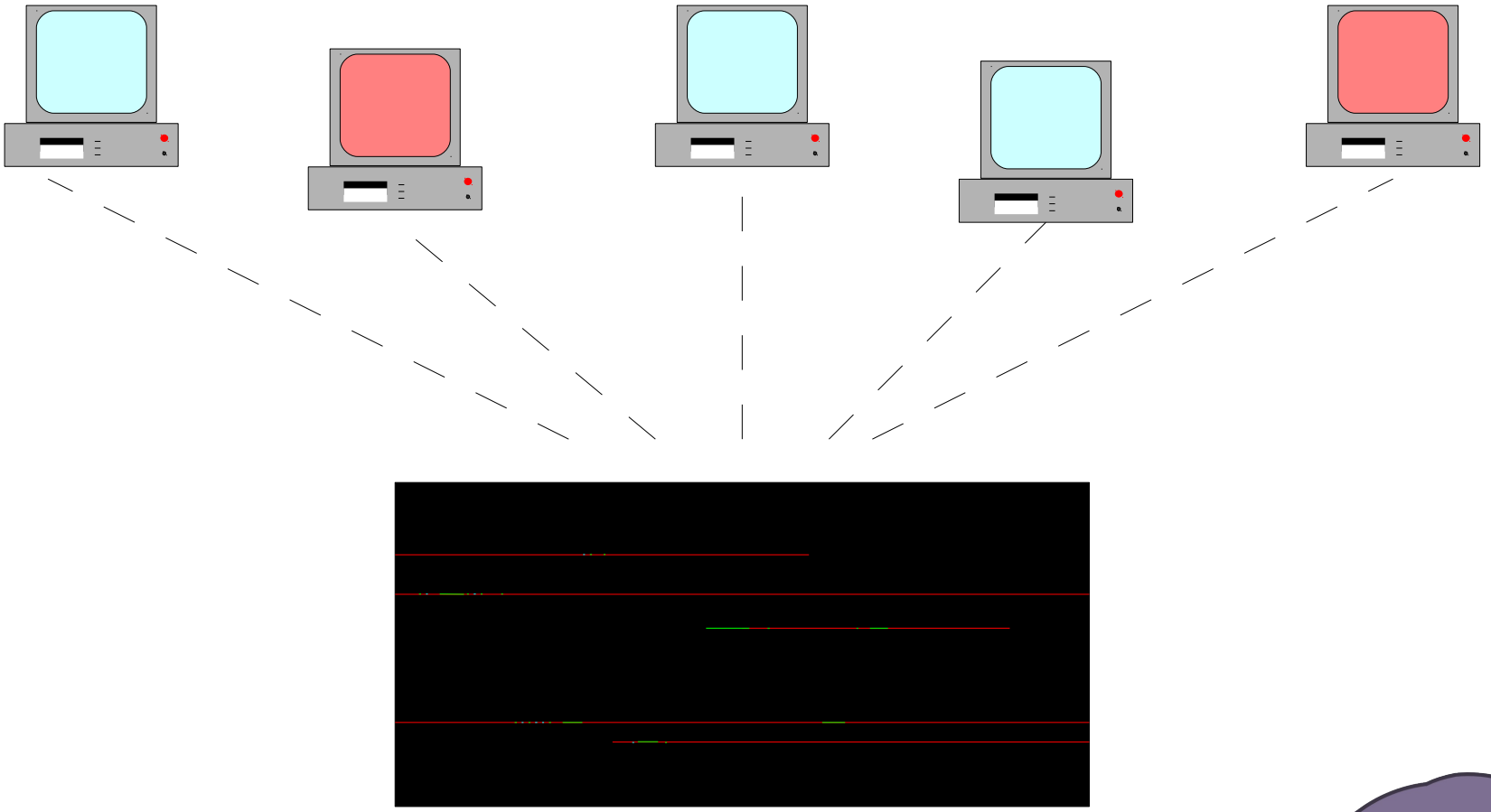
Masoume Jabbarifar
Michel Dagenais
Robert Roy

DORSAL
8 Dec 2010
École Polytechnique, Montreal

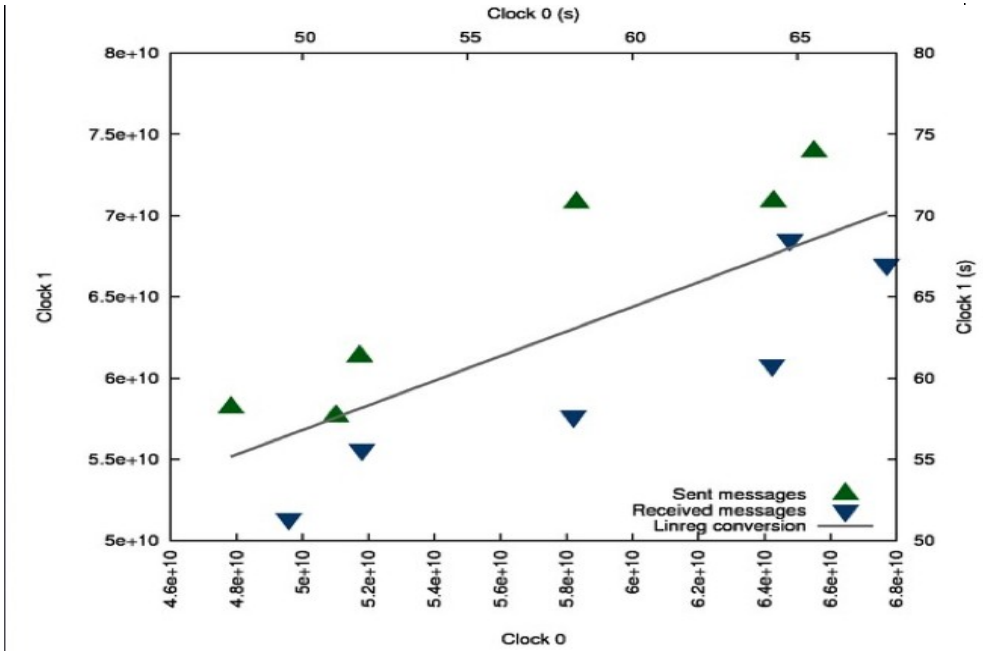
Content

- **Why synchronization**
- **Synchronization methods**
- **Problem and Goals**
- **Architecture**
- **Synchronization optimization**
- **Mammoth cluster**
- **NS2 Simulation**
- **Results**
- **Streaming trace synchronization**
- **Challenges**
- **References**

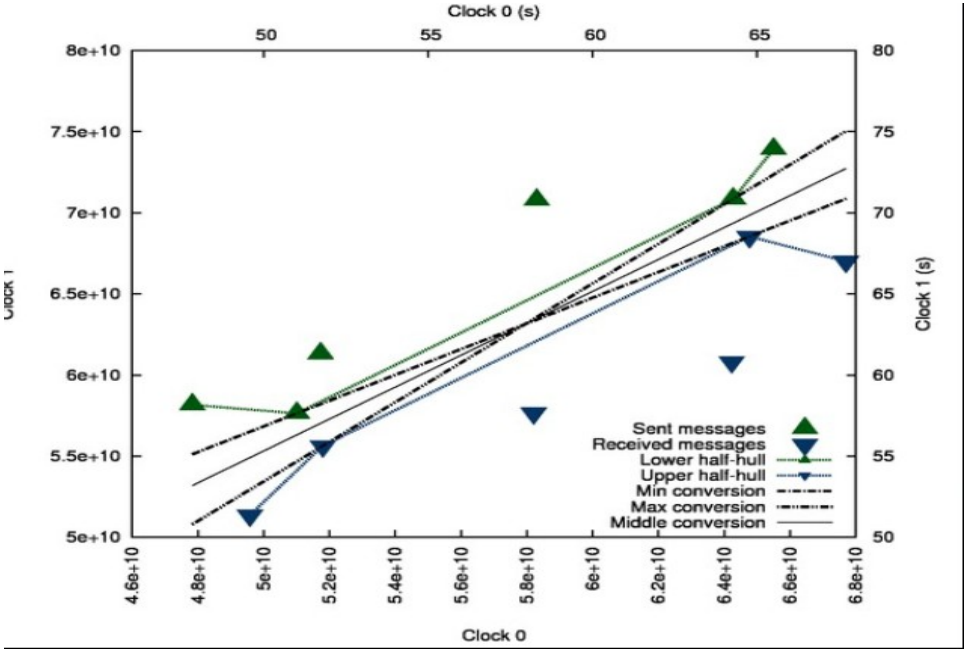
Why synchronization?



Synchronization Methods



Linear regression



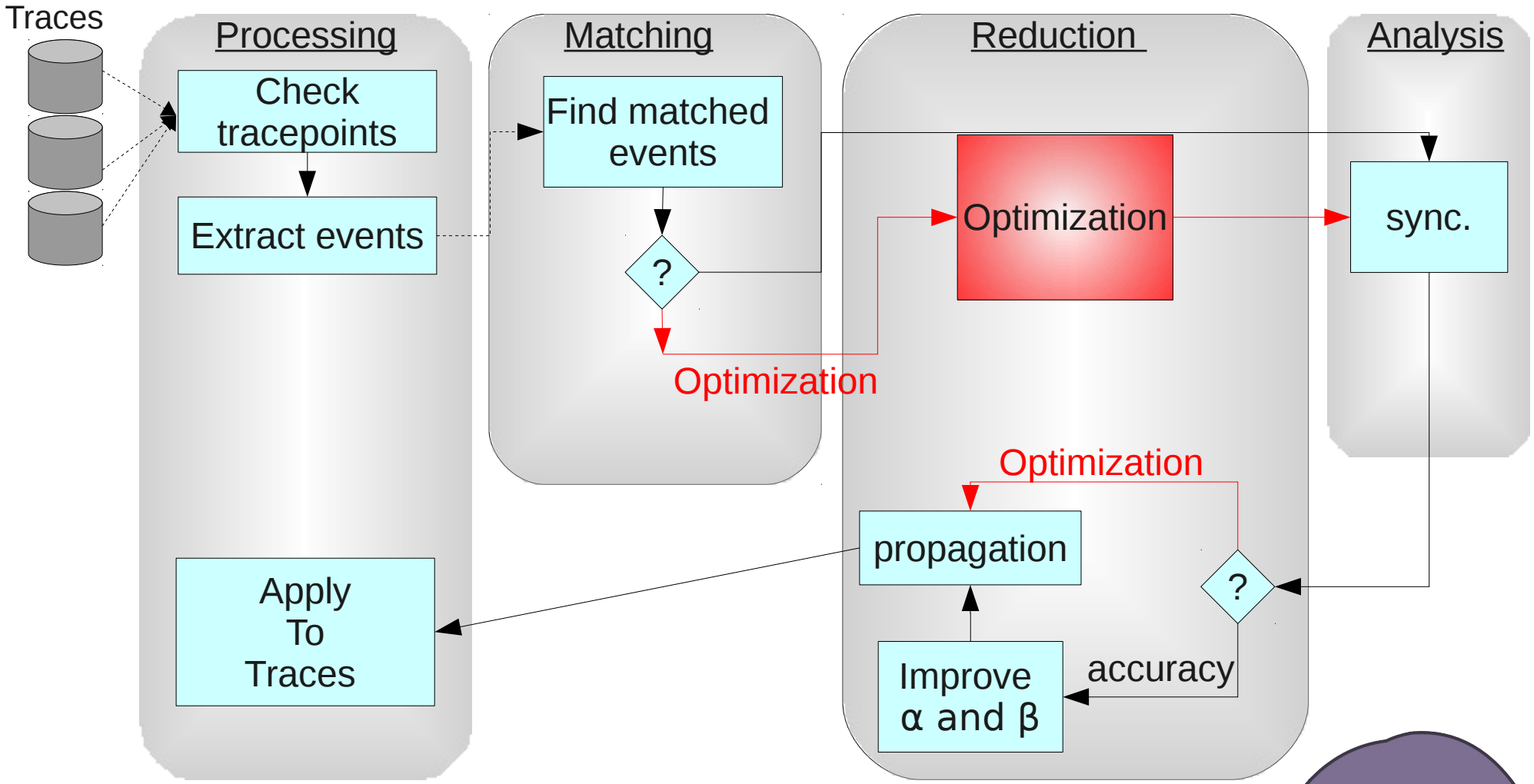
Convex Hull

$$clock1 = \alpha + \beta clock0$$

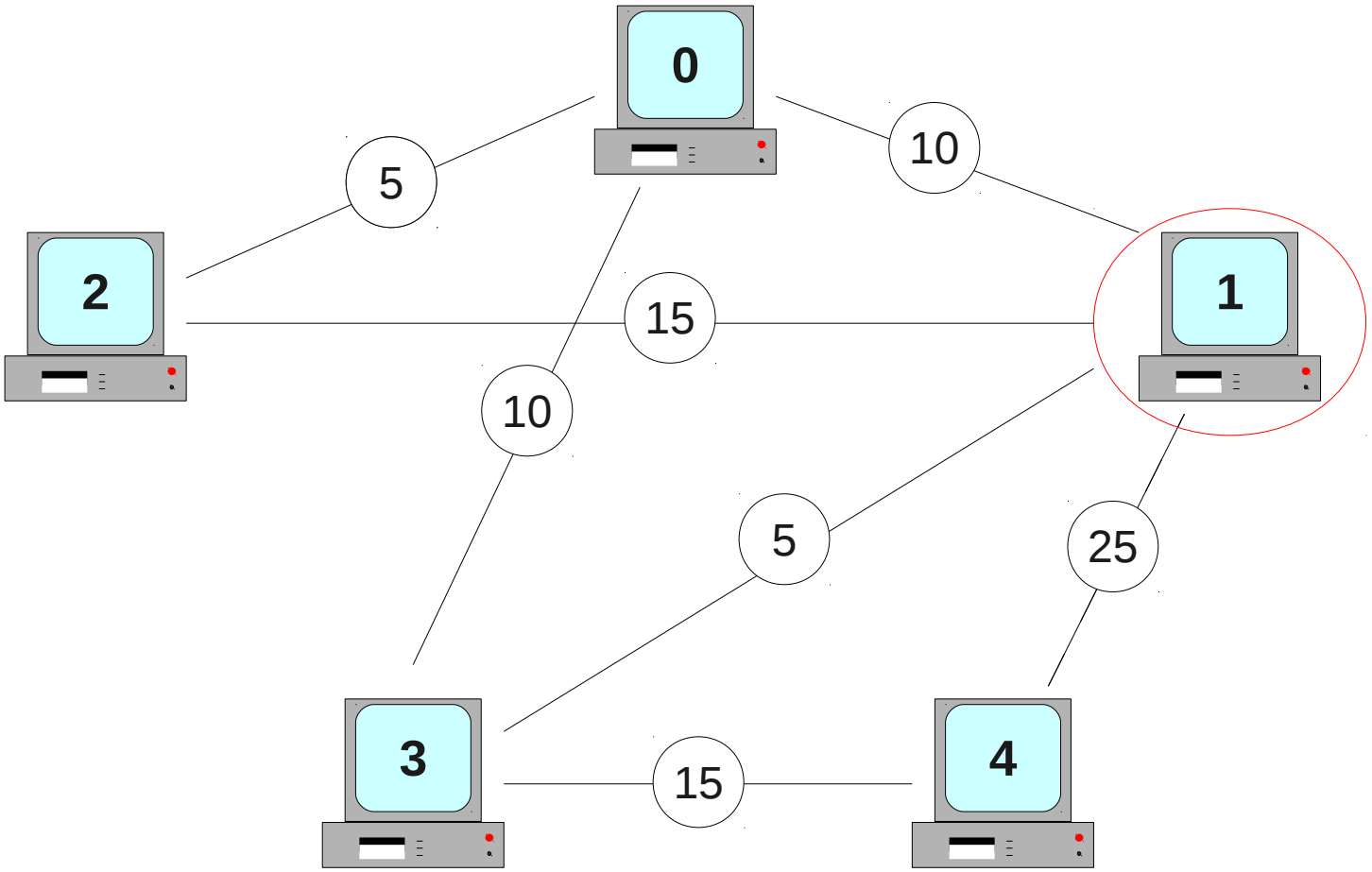
Problem and Goals?

- Total synchronization time increases with the number of nodes and packet exchanges in the network!
  - » For example, with 21 nodes and about 200,000 packets, synchronization takes 20 minutes.
- Optimization Goals:
  - Save synchronization time
  - Keep total accuracy

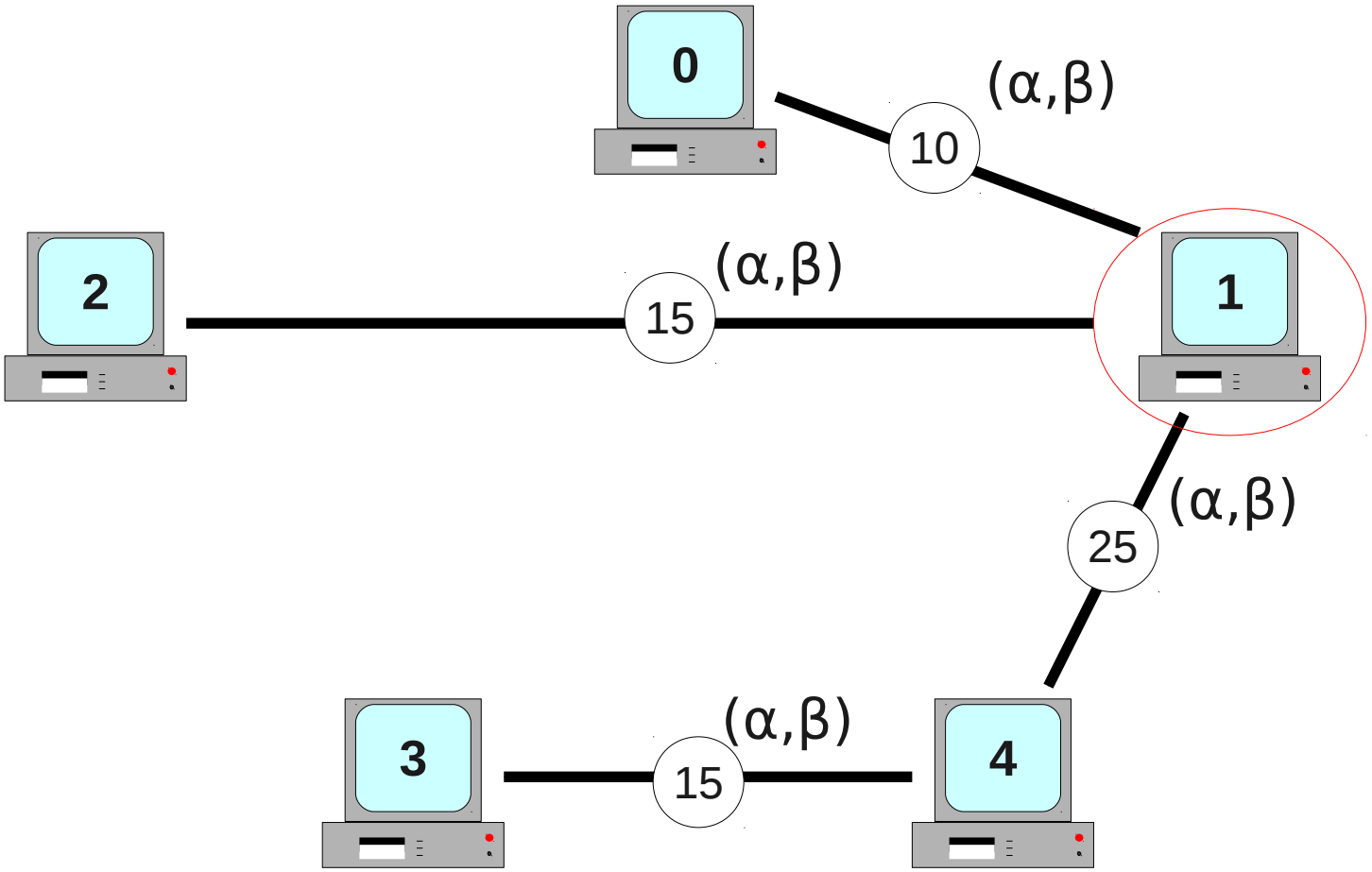
Architecture



Synchronization



Optimized Synchronization



Accuracy Parameters

- Distance
- Quality of network path
- Network latency
- ...

Two Explicit Parameters:

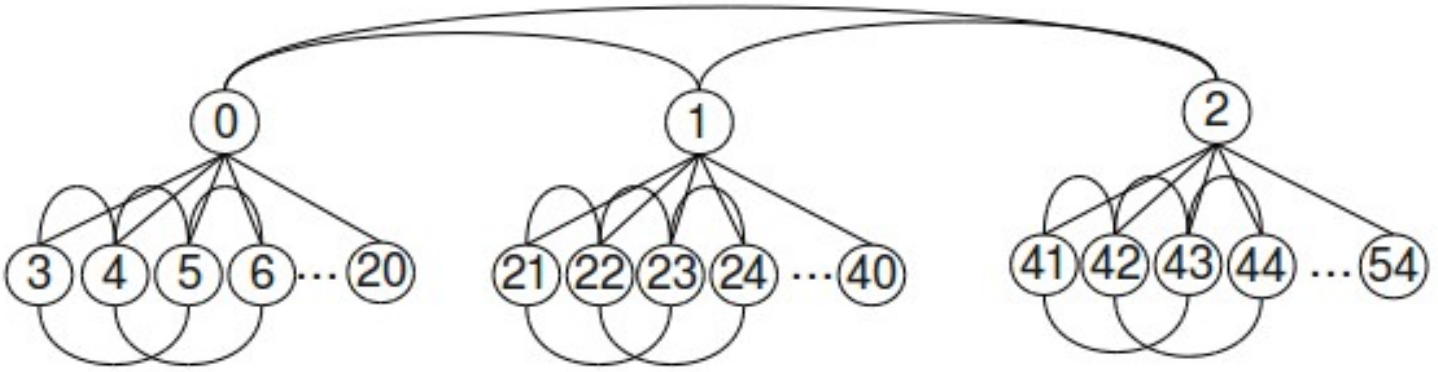
- The number of exchanged packet
- The number of hops to the Reference Node

MAMMOTH Cluster

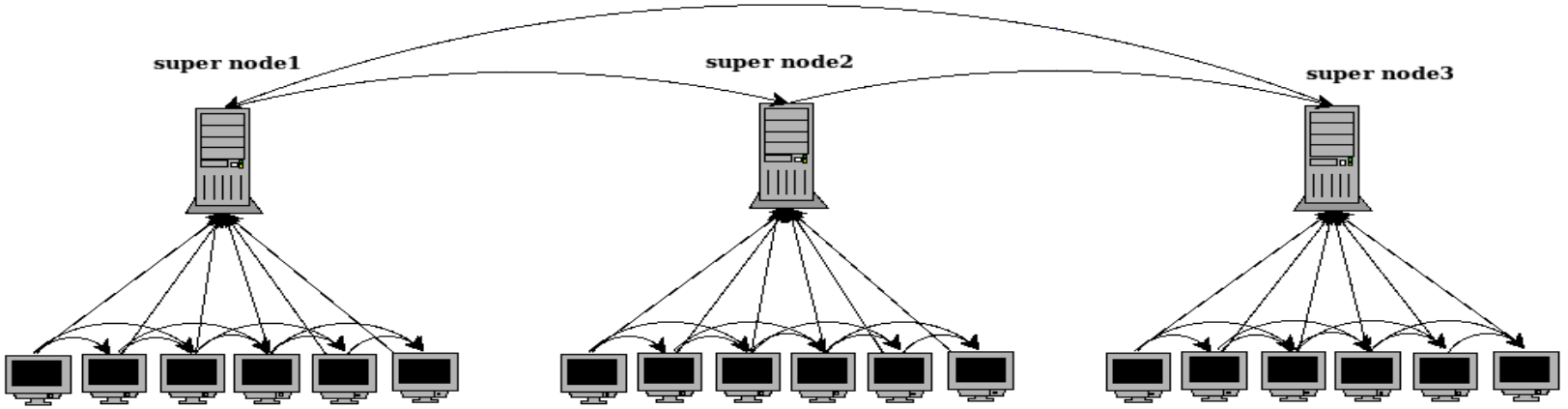
Mammoth is a very large Linux cluster located in Sherbrooke University

It contains two partitions:

- Serial: Pentium 4 computers connected by Gigabit network
- Parallel: Opteron connected by an Infiniband network



NS2 Simulation



Demo

Result (1/2)

No. of Nodes	Total No. of Packets	Previous Sync. Time	Optimized Sync. Time	Saved Time	Percentage
4	1437	8.669469	6.042749	2.5 s	30%
5	2098	13.393313	7.94.772	5.5 s	40%
6	13044	79.606987	69.066550	10.5 s	13%
21	173985	19.5 min	15.5 min	4 min	20%

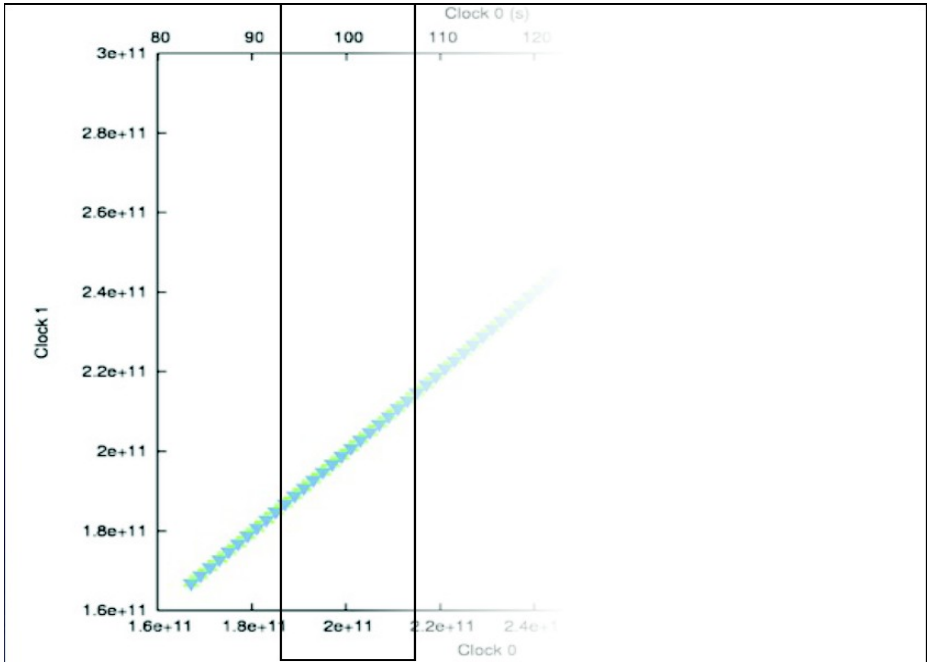
Result (2/2)

- 10 to 40% time optimization depends on:

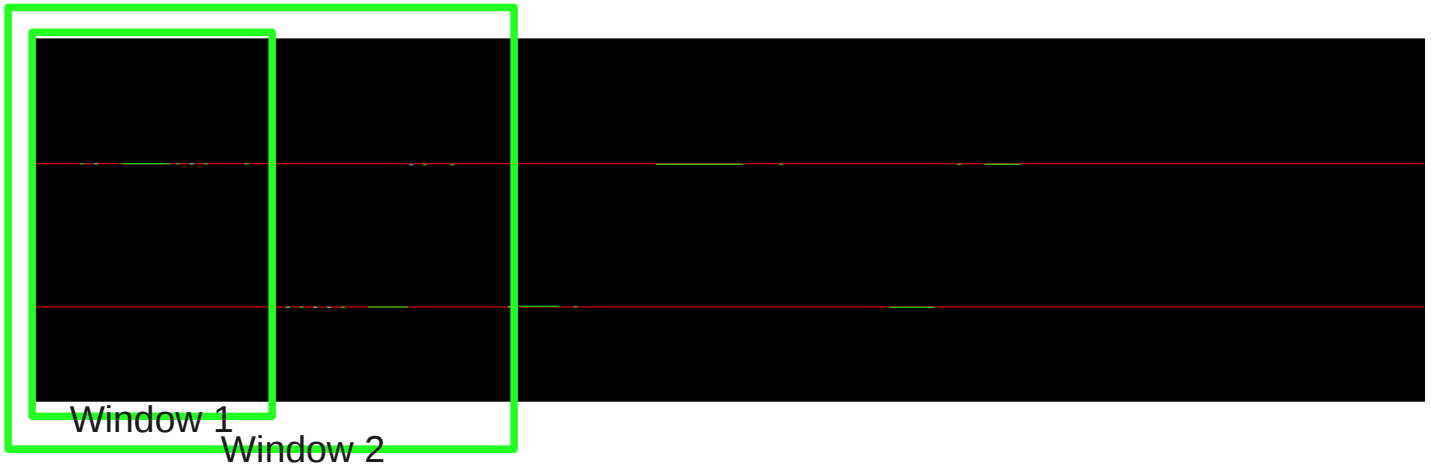
1. Number of removed links
2. Number of packets in removed links

Streaming Trace Synchronization

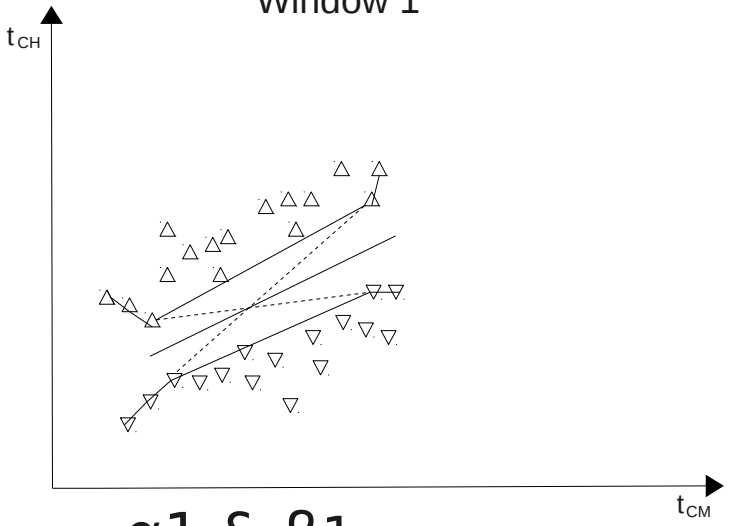
- Sliding window
  - Combine with convex hull



Streaming Trace Synchronization

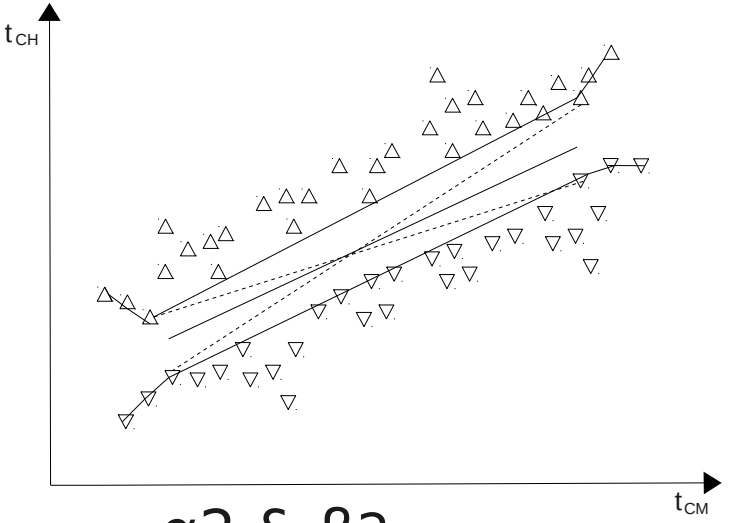


Window 1



α1 & β1

Window 2



α2 & β2

...

Streaming history

- Keep relevant information from previous window:

  1. No need to repeat processing and matching of packets.

  2. Save and reuse previous points located on the convex hull.

Challenges in streaming mode

- Some nodes may be unconnected
- Round Trip Time is needed for Convex-hull and there is always delay to send Acks
- Buffering
- ...

Conclusion and Future work

- Integration of streaming synchronization with Lttv
- Optimizations
- Optimizing streaming synchronization for multiple nodes
- Simulations
- Testing on real hardware environment

References

[1]	B. Poirier, R. Roy and M. Dagenais, “Accurate offline synchronization of distributed traces using kernel-level events, 2010.
[2]	J. H. Deschenes, M. Desnoyers and M. Dagenais. “Tracing Time Operating System State Determination,” The Open Software Engineering Journal, vol. 2, 2008, pp. 40-44.
[3]	A. D. Ksehmkalyani and M. Singhal, “Logical time,” in Distributed Computing: Principles, Algorithms, and Systems, 1st ed., USA: Cambridge University Press, 2008, pp. 50-84.
[4]	H. Khlifi and J. C. Gregorie, " Low-complexity offline and online clock skew estimation and removal," The International Journal of Computer and Telecommunications Networking, vol. 50, no. 11, pp. 1872-1884, 2006.
[5]	L. Chai, Q. Gao and D. K. Panda, “Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System,” Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid, Rio De Janeiro, Brazil, 2007, pp. 471-478.
[6]	J. M. Jezequel and C. Jard, “Building a global clock for observing computations in distributed memory parallel computers,” Concurrency: Practice and Experience, vol 2, no. 1, 1996, pp. 71-89
[7]	E. Betti, M. Cesati, R Gioiosa and F. Piermaria, “A global operating system for HPC clusters,” IEEE International Conference on Cluster Computing and Workshops, 2009.
[8]	R. Sirdey and F. Maurice, “A linear programming approach to highly precise clock synchronization over a packet network,” 4OR: A Quarterly Journal of Operations Research, vol. 6, no. 4, 2008, pp. 393-401.
[9]	C. N. Keltcher, K. J. McGrath, A. Ahmed, and P. Conway, “The amd opteron processor for multiprocessor servers,” IEEE Micro, vol. 23, no. 2, 2003, pp. 66–76.