Multi-level, Multi-core Distributed Trace Synchronization

Masoume Jabbarifar

Masoume.jabbarifar@polymtl.ca

Supervisor: Michel Dagenais

DORSAL

11 May 2011

Outline

- **Optimization on Offline Synchronization**
  - **Convex-Hull**
  - **Architecture**
  - **Results**
- **Online Synchronization**
  - **Interval based Aposteriori Synchronization**
  - **Sliding Window based Synchronization**
  - **Incremental Online Synchronization**
- **Conclusion**
- **References**

Synchronization Algorithm

Convex-Hull

1) Sent and Received sets

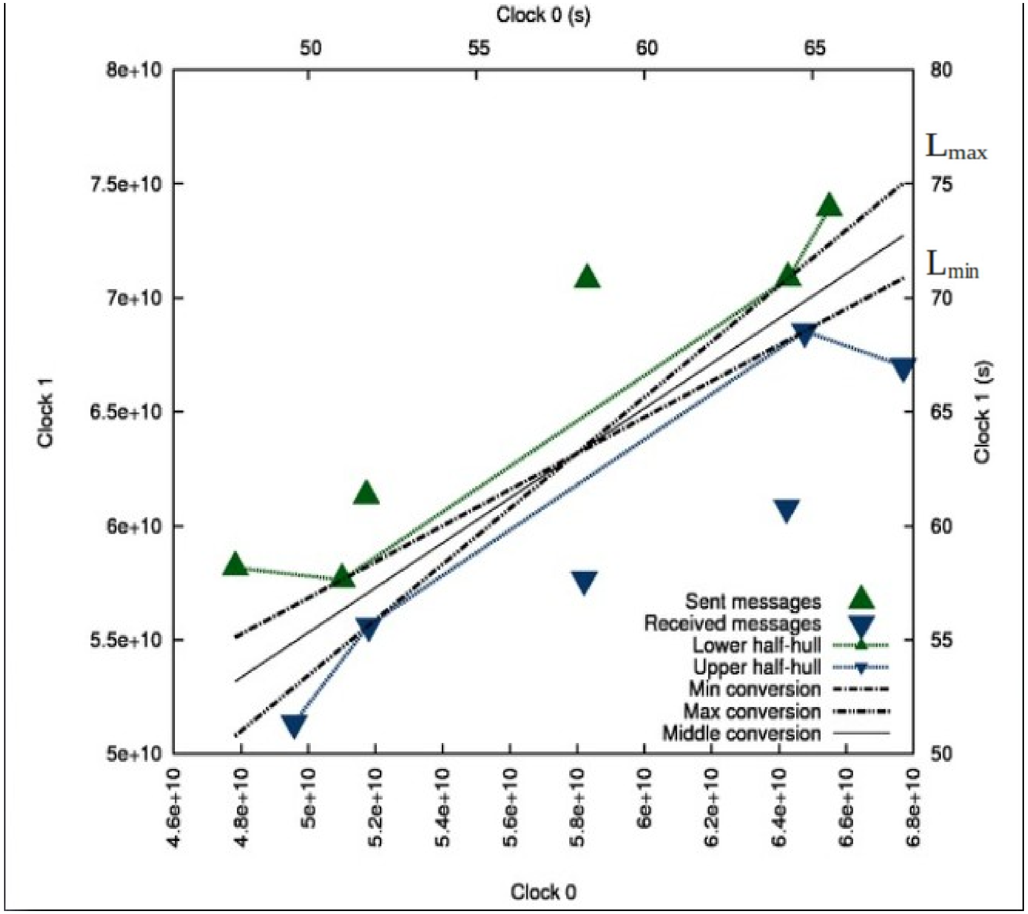
- ➢ Guarantee no message inversion

2) Two lines with Max & Min slop

$$L_{max}(t_A) = a_1^{max} t_A + b_0^{min}$$

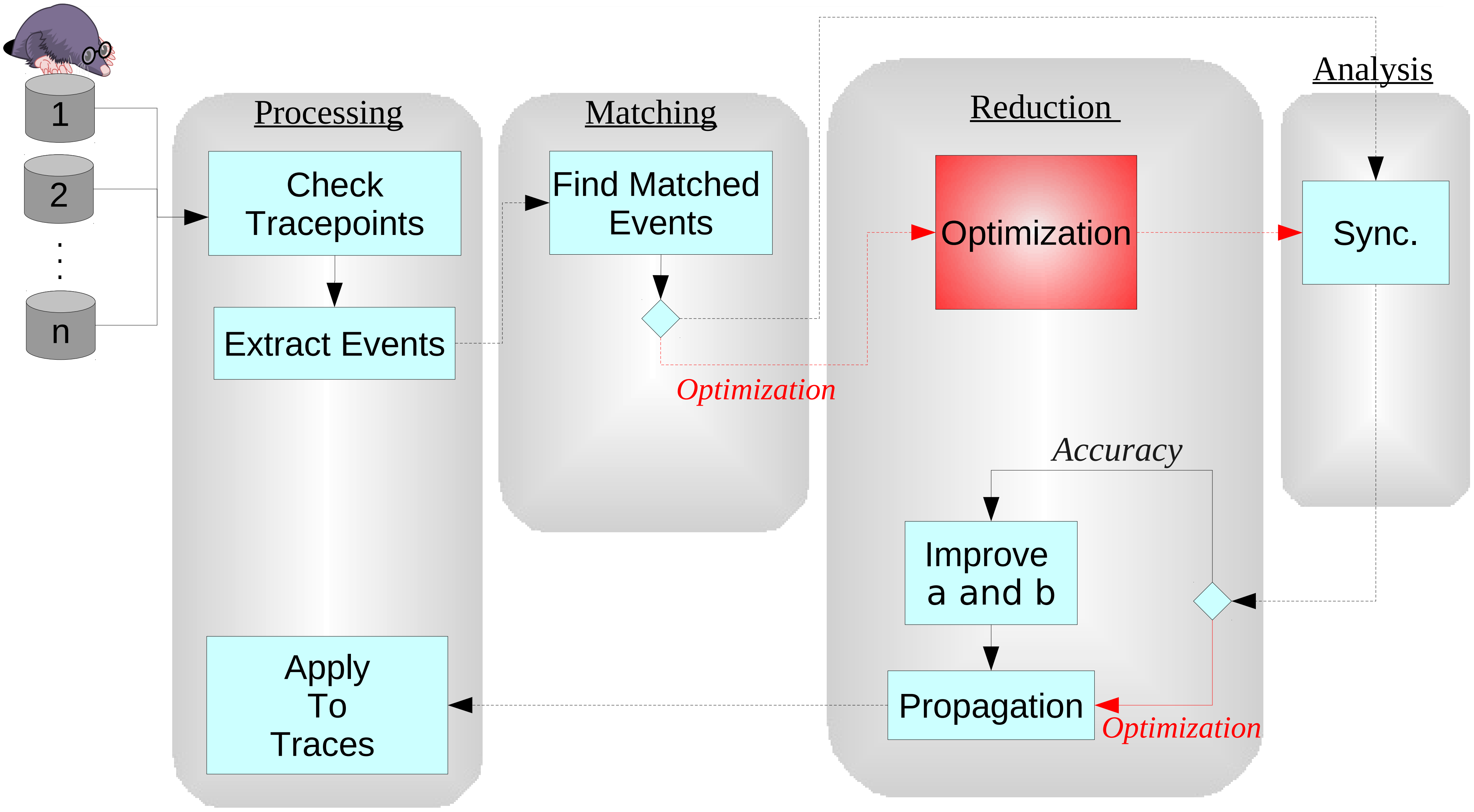
$$L_{min}(t_A) = a_1^{min} t_A + b_0^{max}$$

$$Accuracy = a_1^{max} - a_1^{min}$$



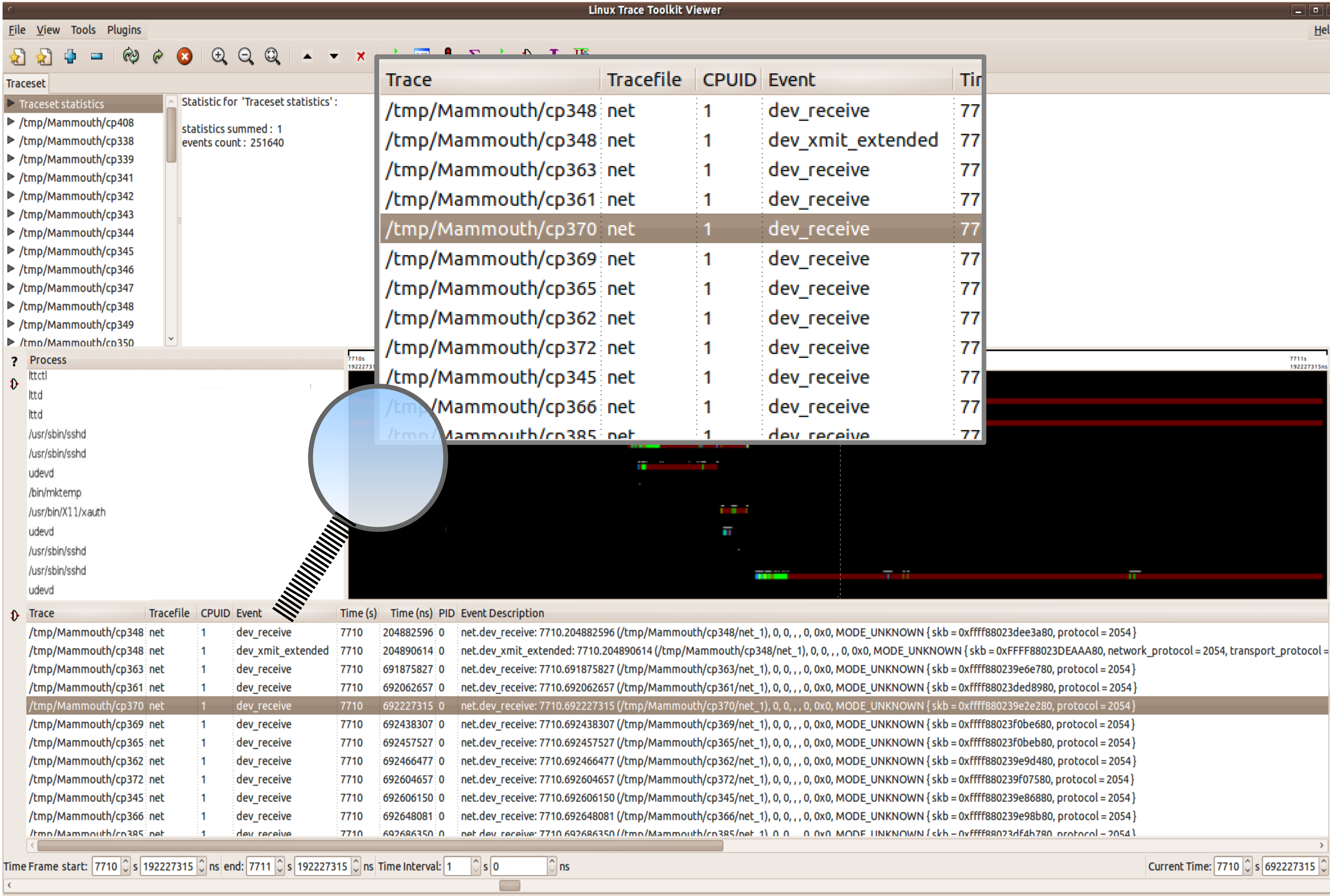
3)The bisector of the angle formed by these two lines

Architecture

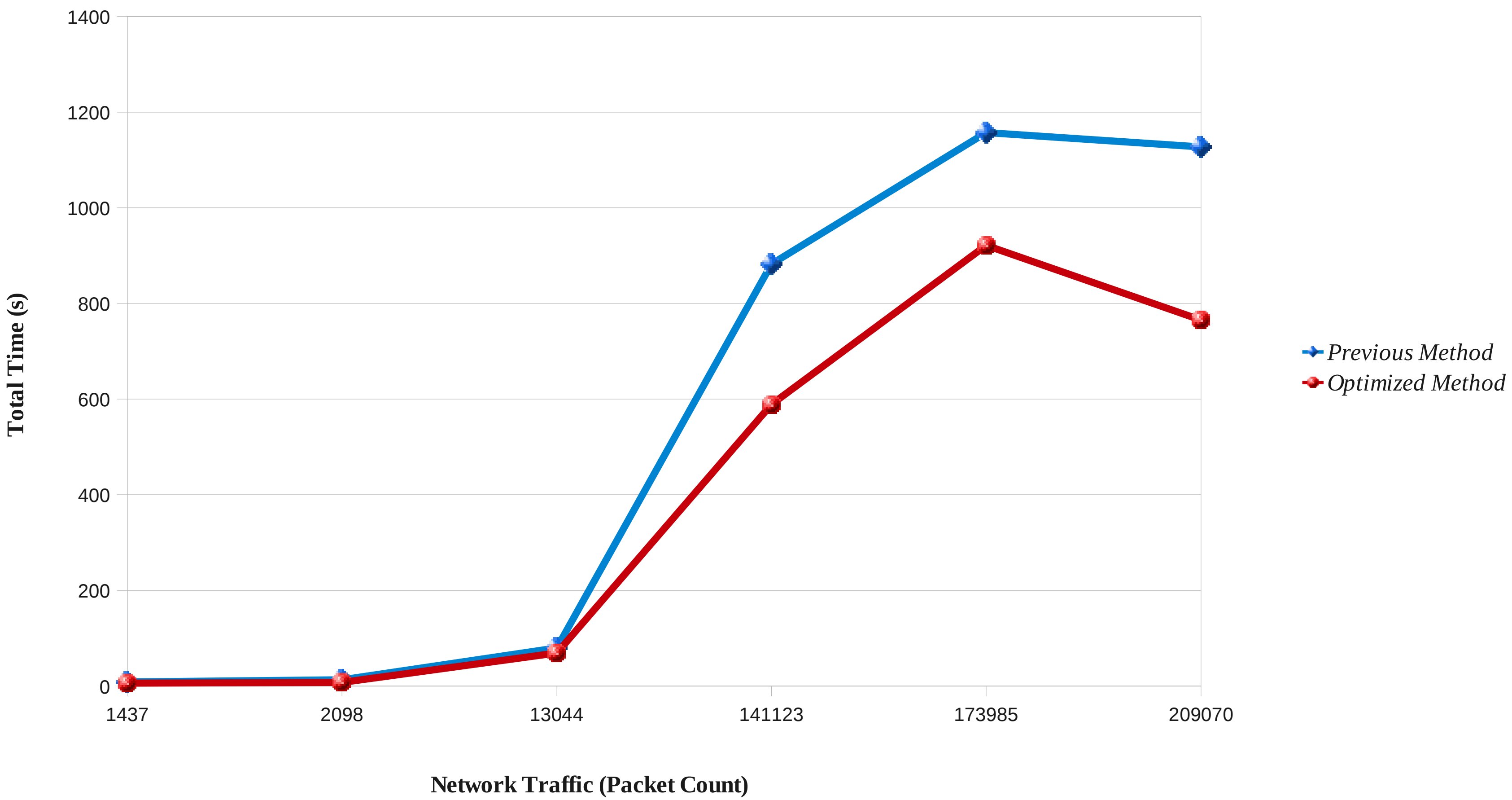


Network Features

1) Physical distance
2) Quality of network path
3) Network latency
4) Delays
5) Hop count
6) Network traffic


Linux Trace Toolkit Viewer
File View Tools Plugins
Help
Traceset
Traceset statistics
/tmp/Mammouth/cp408
/tmp/Mammouth/cp338
/tmp/Mammouth/cp339
/tmp/Mammouth/cp341
/tmp/Mammouth/cp342
/tmp/Mammouth/cp343
/tmp/Mammouth/cp344
/tmp/Mammouth/cp345
/tmp/Mammouth/cp346
/tmp/Mammouth/cp347
/tmp/Mammouth/cp348
/tmp/Mammouth/cp349
Statistic for 'Traceset statistics' :
statistics summed : 1
events count : 251640
Trace Tracefile CPUID Event
/tmp/Mammouth/cp348 net 1 dev_receive 77
/tmp/Mammouth/cp348 net 1 dev_xmit_extended 77
/tmp/Mammouth/cp363 net 1 dev_receive 77
/tmp/Mammouth/cp361 net 1 dev_receive 77
/tmp/Mammouth/cp370 net 1 dev_receive 77
/tmp/Mammouth/cp369 net 1 dev_receive 77
/tmp/Mammouth/cp365 net 1 dev_receive 77
/tmp/Mammouth/cp362 net 1 dev_receive 77
/tmp/Mammouth/cp372 net 1 dev_receive 77
/tmp/Mammouth/cp345 net 1 dev_receive 77
/tmp/Mammouth/cp366 net 1 dev_receive 77
Process
lttctl
lttd
lttd
/usr/sbin/sshd
/usr/sbin/sshd
udevd
/bin/mktemp
/usr/bin/X11/xauth
udevd
/usr/sbin/sshd
/usr/sbin/sshd
udevd
Trace Tracefile CPUID Event Time (s) Time (ns) PID Event Description
/tmp/Mammouth/cp348 net 1 dev_receive 7710 204882596 0 net.dev_receive: 7710.204882596 (/tmp/Mammouth/cp348/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff88023dee3a80, protocol = 2054 }
/tmp/Mammouth/cp348 net 1 dev_xmit_extended 7710 204890614 0 net.dev_xmit_extended: 7710.204890614 (/tmp/Mammouth/cp348/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xFFFF88023DEAAA80, network_protocol = 2054, transport_protocol =
/tmp/Mammouth/cp363 net 1 dev_receive 7710 691875827 0 net.dev_receive: 7710.691875827 (/tmp/Mammouth/cp363/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff880239e6e780, protocol = 2054 }
/tmp/Mammouth/cp361 net 1 dev_receive 7710 692062657 0 net.dev_receive: 7710.692062657 (/tmp/Mammouth/cp361/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff88023ded8980, protocol = 2054 }
/tmp/Mammouth/cp370 net 1 dev_receive 7710 692227315 0 net.dev_receive: 7710.692227315 (/tmp/Mammouth/cp370/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff880239e2e280, protocol = 2054 }
/tmp/Mammouth/cp369 net 1 dev_receive 7710 692438307 0 net.dev_receive: 7710.692438307 (/tmp/Mammouth/cp369/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff88023f0be680, protocol = 2054 }
/tmp/Mammouth/cp365 net 1 dev_receive 7710 692457527 0 net.dev_receive: 7710.692457527 (/tmp/Mammouth/cp365/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff88023f0beb80, protocol = 2054 }
/tmp/Mammouth/cp362 net 1 dev_receive 7710 692466477 0 net.dev_receive: 7710.692466477 (/tmp/Mammouth/cp362/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff880239e9d480, protocol = 2054 }
/tmp/Mammouth/cp372 net 1 dev_receive 7710 692604657 0 net.dev_receive: 7710.692604657 (/tmp/Mammouth/cp372/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff880239f07580, protocol = 2054 }
/tmp/Mammouth/cp345 net 1 dev_receive 7710 692606150 0 net.dev_receive: 7710.692606150 (/tmp/Mammouth/cp345/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff880239e86880, protocol = 2054 }
/tmp/Mammouth/cp366 net 1 dev_receive 7710 692648081 0 net.dev_receive: 7710.692648081 (/tmp/Mammouth/cp366/net_1), 0, 0, , , 0, 0x0, MODE_UNKNOWN { skb = 0xffff880239e98b80, protocol = 2054 }
Time Frame start: 7710 s 192227315 ns end: 7711 s 192227315 ns Time Interval: 1 s 0 ns
Current Time: 7710 s 692227315 ns

Result of NS2 (1/2)



Result of NS2 (2/2)

No. of Nodes	Total No. of Packets	Previous Sync. Time	Optimized Sync. Time	Saved Time (s)	Percentage
4	1437	8.67	6.04	2.5	30 %
5	2098	13.39	7.94	5.5	40 %
6	13044	79.60	69.06	10.5	13 %
16	209070	1127.28	765.22	362.06	32 %
19	141123	882.43	588.89	293.54	33 %
21	173985	1157.051	921.3	235.75	20 %
Average					**28 %**

Outline

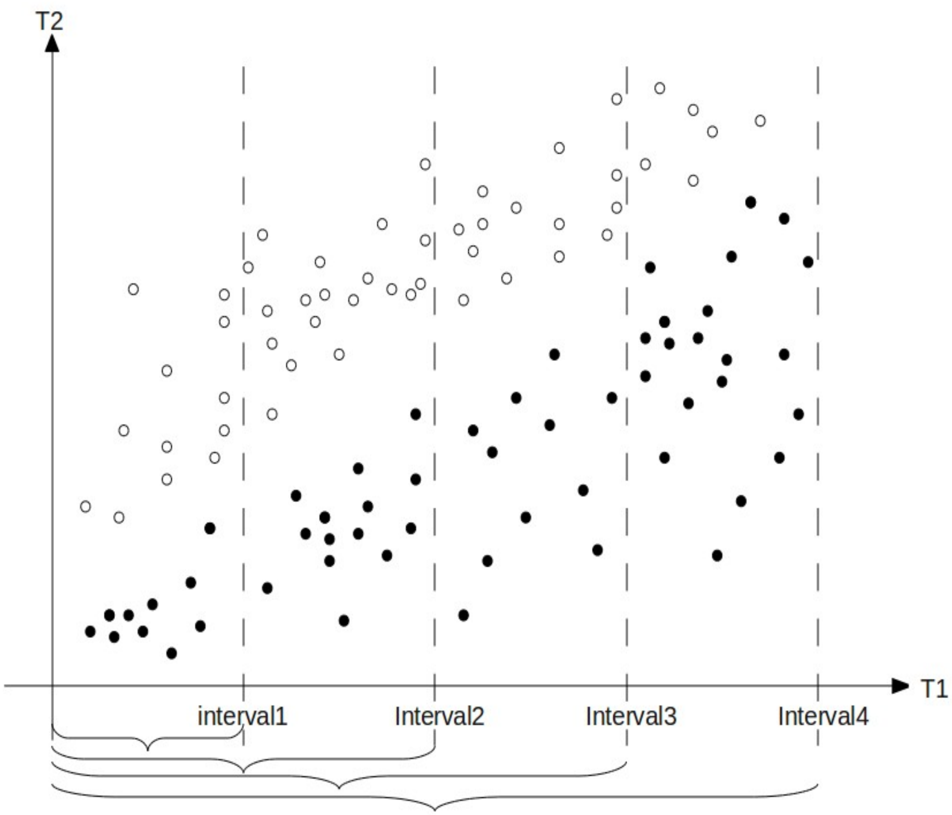
- Optimization on Offline Synchronization
  - Convex-Hull
  - Architecture
  - Results
- **Online Synchronization**
  - **Interval based Aposteriori Synchronization**
  - **Sliding Window based Synchronization**
  - **Incremental Online Synchronization**
- Conclusion
- Reference

Interval based Aposteriori Synchronization

- Incremental interval
- Save and reuse previous points

- Analysis on the whole data from the start point of tracing

- No need to repeat processing and matching of packets



Interval based Aposteriori Synchronization

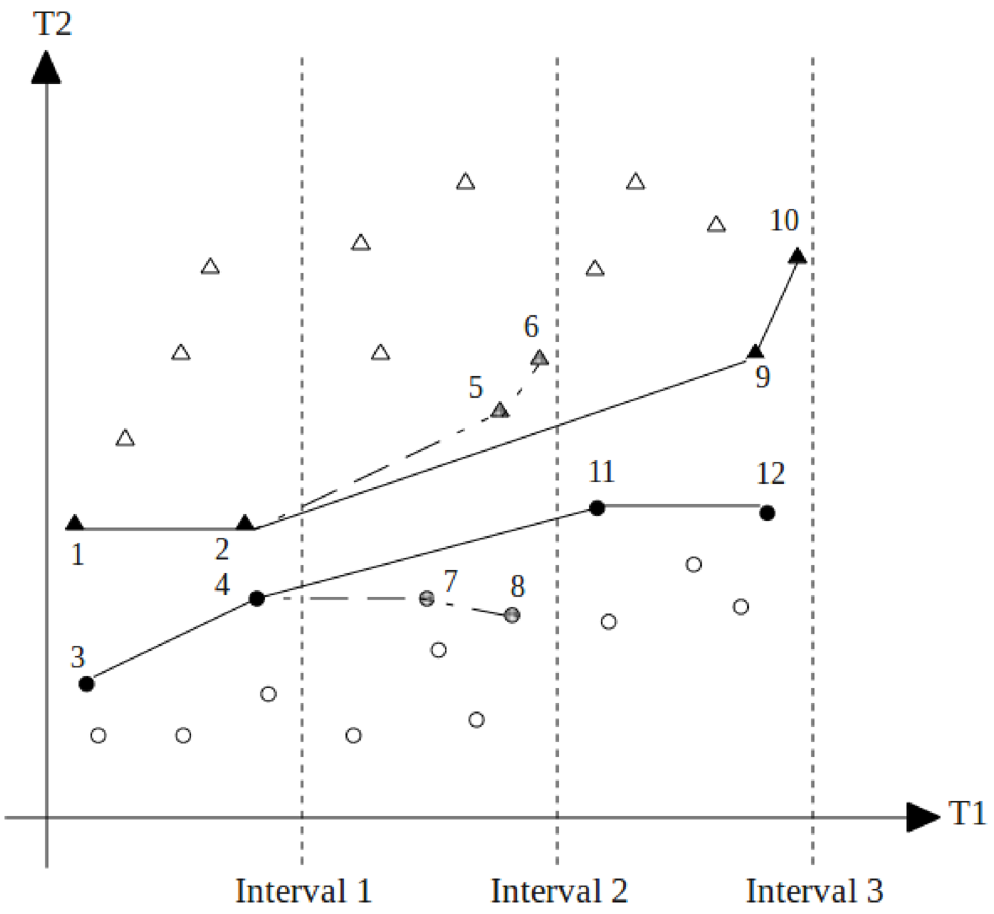
- **The advantage:**
  - The highest level of accuracy
- **The disadvantages:**
  - Scalability
- **Optimization:**
  - Consider particular no. of previous intervals (e.g. 5 intervals)

LTTV Integration Challenges

- Add /remove new/old node at the entrance/leave time
- Gather trace files
- Synchronization delays (Network, Algorithm)
- Buffering

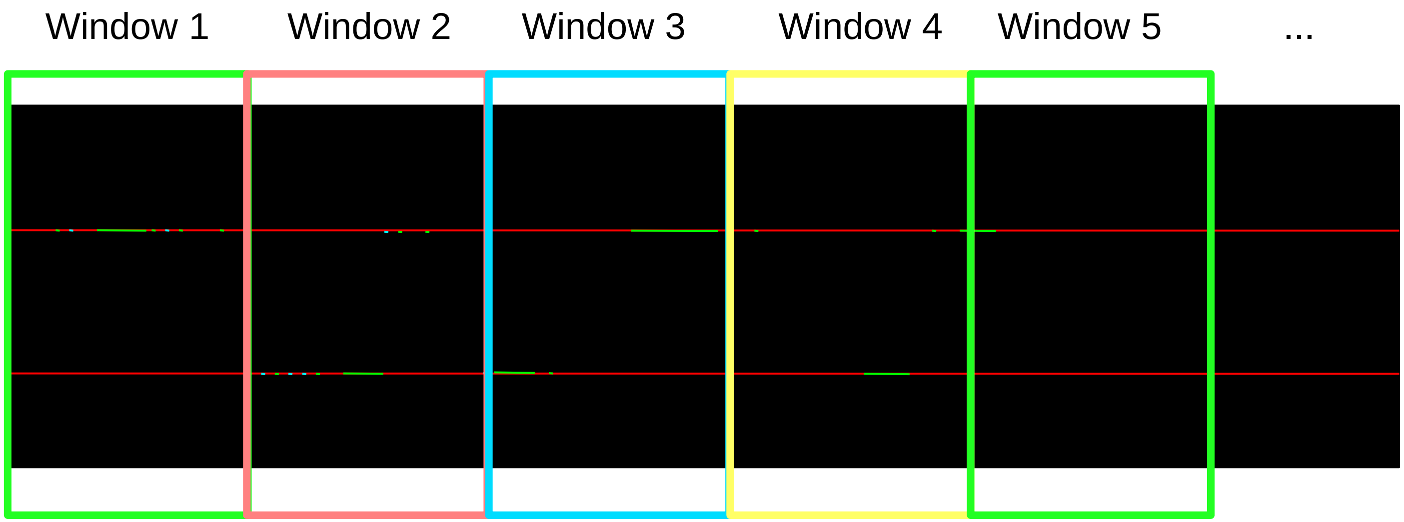
Sliding Window based Synchronization

- L: size of window (static interval)
- Accurate packet is replaced as soon as detected



Sliding Window based Synchronization

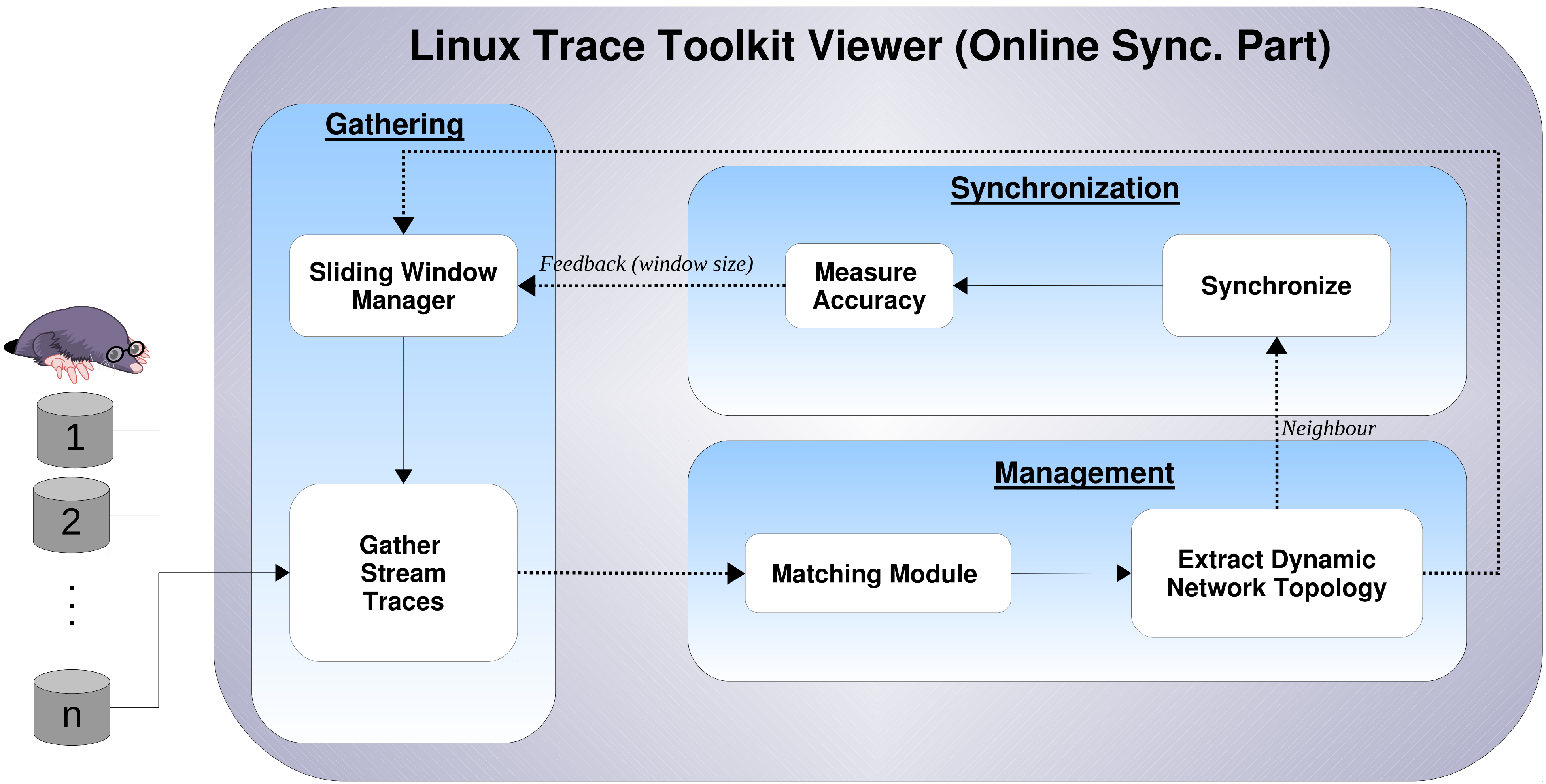
- **The advantages:**
  - Guarantee high accuracy all the time
  - Improve accuracy over time
  - No buffering



Incremental Online Synchronization

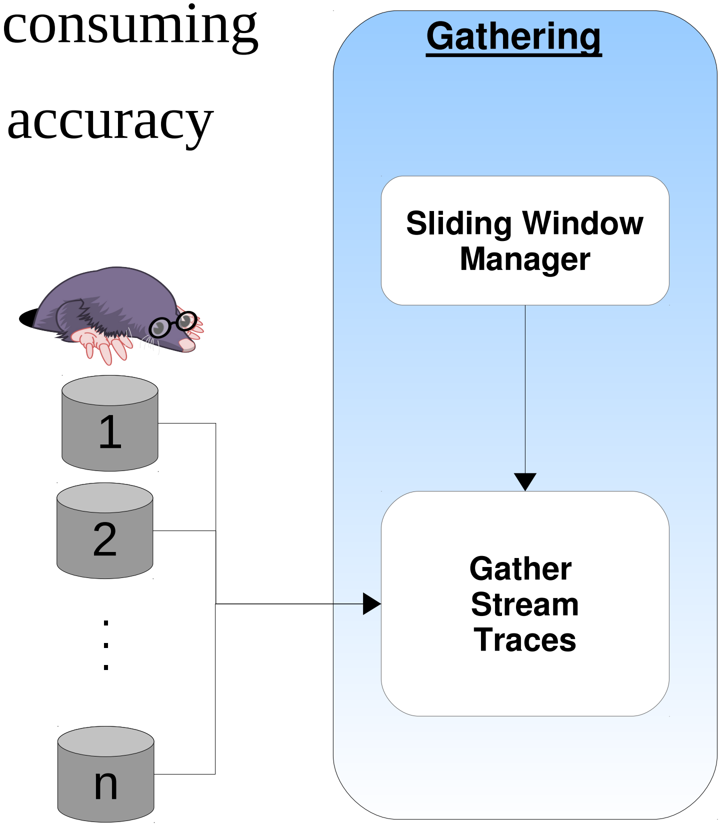
- Self-Managing Method
- Optimize performance of the synchronization
- Dynamic window size

Architecture



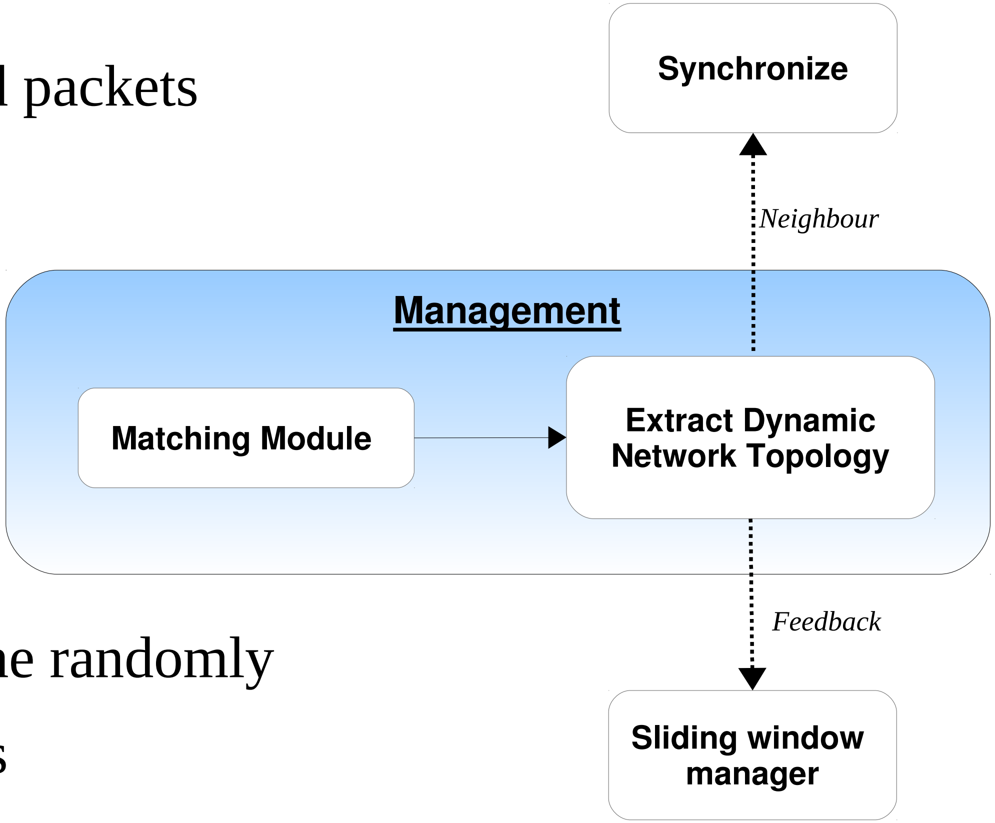
Gathering Component

- **Sliding window manager:**
  - Wide window = high accuracy & time consuming
  - Narrow window = performance & low accuracy
  - Network situation = dynamic/stable
- **Gathering Stream Traces:**
  - Network traffic load
  - Streaming latency



Management Component

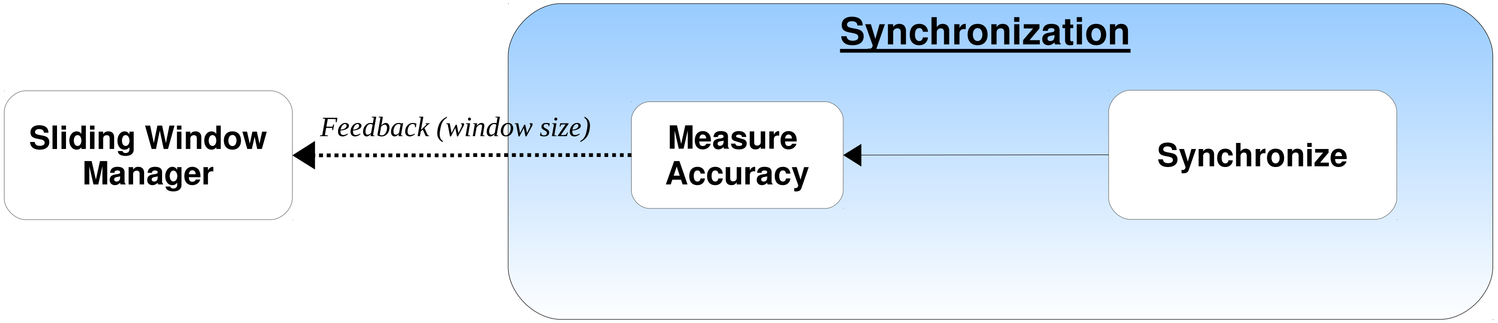
- **Match Packets:**
  - Manage unknown received packets
- **Extract Dynamic Network:**
  - Extract set of nodes :
    - Communications
  - Select neighbors one by one randomly
  - Feedback network changes



Synchronization Component

- **Measure Accuracy:**
  - Balance speed and accuracy
  - Change window size

- **Synchronize**
  - Synchronize all nodes



Conclusion and Future Work

- Accurate online synchronization
- Scalable online synchronization
- Increamental online synchronization for large-scale dynamic systems

References (1)

[1] B. Poirier, R. Roy, and M. Dagenais, “Accurate Offline Synchronization of Distributed Traces Using Kernel-level Events," Operating Systems Review, vol. 44, 2010, pp. 75-87.

[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. Duda, G. Harrus, Y. Haddad, and G. Bernard, “Estimation global time in distributed system," In proceeding 7th Int. Conf. on Distributed Computing Systems, Berlin, volume 18, 1987.

[4] S.B. Moon, P. Skelly, D. Towsley, “Estimation and Removal of Clock Skew from Network Delay Measurements," in: INFOCOM, 1999.

[5] 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.

[6] 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.

[7] M. Bligh, M. Desnoyers, and R. Schultz, “Linux kernel debugging on google-sized clusters," In Proceedings of the Linux Symposium, 2007.

[8] M. Desnoyers, “Low-Impact Operationg System Tracing," PhD thesis, Ecole Polytechnique de Montreal, 2009.

[9] 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.

[10] B. Scheuermann, W. Kiess, M.Roos, F. Jarre, and M. Mauve, “On the time synchronization of distributed log files in networks with local broadcast media," IEEE/ACM Transactions on Networking, vol. 17 no.2, 2009, pp. 431-444.

[11] J. Jezequel, and C. Jard, “Building a global clock for observing computations in distributed memory parallel computers," Concurrency: Practice and Experience, vol. 8 no.1, 1996.

References (2)

[12] H. Marouani, and M.R. Dagenais, “Internal Clock Drift Estimation in Computer Clusters," Journal of Computer Systems, Networks, and Communications, vol.2008 no. 1, 2008, pp. 1-7.

[13] L.M. He, “Time Synchronization Based on Spanning Tree for Wireless Sensor Networks," 4th International Conference on Wireless Communications, Networking and mobile Computing, Dalian, 2008, pp. 1-4.

[14] Mammoth project available at “https://rqchp.ca/?mod=cms&pageId=566&lang=EN," Sep. 2010.

[15] 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.

[16] 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

[17] 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.

[18] 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.

[19] M. Papakipos, “High-Productivity Software Development for Multi-Core Processors,” 2007. [Online]. Available: http://download.microsoft.com/download/d/f/6/df6accd5-4bf2-4984-8285-f4f23b7b1f37/WinHEC2007_PeakStream.doc [Accessed: 14 April 2010].

[20] NIST Time and frequency from A to Z., February 2011. http://tf.nist.gov/general/glossary.htm.

[21] E. Clement and M. Dagenais, “Trace synchronization in distributed networks,” Journal of computer system, Network, and Communication, 2009.

[22] P. Ashton, “Algorithms for off-line clock synchronization,” Technical report, University of Canterbury, Department of Computer Science, Dec. 1995.

References (3)

[23] J. Doleschal, A. Kn¨pfer, M. S. M¨ller, and W. E. Nagel, "Internal timer synchronization for parallel event tracing," In Proceedings of the 15th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, pages 202–209, Berlin, Heidelberg, 2008. Springer-Verlag.

[24] K. Berket, R. Koch, L.E. Moser, P.M. Melliar-Smith, "Timestamp acknowledgements for determining message stability," in: Proceedings of the Second International Conference on Parallel and Distributed Computing and Networks, Brisbane, Australia, December 1998.

[25] B. Scheuermann, W. Kiess, M. Roos, F. Jarre, and M. Mauve, "On the time synchronization of distributed log files in networks with local broadcast media," Networking, IEEE/ACM Transactions on, 17(2): 431–444, April 2009.

[26] D. Dolev, N. Lynch, S. Pinter, E. Strark, and W. Weihl, "Reaching approximate agreement in the presence of faults," JACM 33, 3 (July), 499–516, 1986.

[27] J. Joseph, and C. Fellenstein, "Grid Computing," Prentice Hall, 2003.

[28] K. Iwanicki, "Gossip-based dissemination of time," Master's thesis, Warsaw University and Vrije Universiteit Amsterdam (2005).

[29] K. Iwanicki, M. van Steen, S. Voulgaris, "Gossip-based clock synchronization for large decentralized systems," in: Proc. Workshop on Self-Managed Networks, Systems and Services, vol. 3996 of LNCS, Springer, 2006, pp. 28–42.

[30] M. Jelasity, R. Guerraoui, A. -M. Kermarrec, M. van Steen, "The peer sampling service: Experimental evaluation of unstructured gossip-based implementations," in: Proc. ACM/IFIP/USENIX Middleware Conf, vol. 3231 of LNCS, Springer, 2004, pp.79–98.

[31] A. -M. Kermarrec, M. van Steen, "Gossiping in distributed systems," SIGOPS Oper. Syst. Rev. 41 (5) (2007) 2–7.

[32] H. Marouani and M. Dagenais, "Comparing high resolution timestamps in computer clusters," IEEE, 2005.

[33] D. Salyers, A. Striegel, C. Poellabauer, "A Light Weight Method for Maintaining Clock Synchronization for Networked Systems," IEEE, 2008.

References (4)

[34] G. Coulouris, J. Dollimore, T. Kindberg, “ Distributed Systems concepts and design” 4th edition, Addison Wesley, 2005.

[35] A. S. Tanenbaum, M. V. Steen, “Distributed Systems principles and paradigms," 2nd edition, Prentice Hall, 2006.

[36] H.Marouani and M. Dagenais, “Internal Clock Drift Estimation in Computer Cluster," IEEE, 2008.

[37] J. Blunck, M. Desnoyers, and P. -M. Fournier, “Userspace application tracing with markers and tracepoints,” in Proceedings of the 2009 Linux Kongress, Oct. 2009.

[38] M. Desnoyers and M.R. Dagenais, “Deploying LTTng on Exotic Embedded Architectures,” in Embedded Linux Conference 2009, 2009.

[39] Available in http://en.wikipedia.org/wiki/Time_Stamp_Counter in April 2011

[40] M. A. Dietz. “Gathering And Using Time Measurements In Distributed Systems” PhD thesis, Duke University, 1996.

[41] J. Desfossez and M. Dagenais , “Virtual machines traces synchronization” presentation in the Dorsal laboratory, 2010.

[42] M. Jabbarifar, A. S. Sendi, H. Pedram, M. Dehghan and M. Dagenais, “L-SYNC: Larger Degree Clustering Based Time-Synchronisation for Wireless Sensor Network,” Eighth ACIS International Conference on Software Engineering Research, Management and Applications, Montreal, 2010.

[43] M. Jabbarifar, A. S. Sendi, Alireza Sadighian, Naser Ezzati Jivan, and M. Dagenais, “A Reliable and Efficient Time Synchronization Protocol for Heterogeneous Wireless Sensor Network,” Journal of Wireless Sensor Network, vol.2, 2010, pp. 910-918.

[44] M. Jabbarifar, M. Dagenais and R.Roy, “Optimum off-line trace synchronization of computer clusters,” has been sent to HPCS (High Performance Computing Symposium), 2011

References (5)

[45] F. Cristian, “Probabilistic Clock Synchronization,” Distributed Computing, vol. 3, no. 3, pp. 146-158, 1989.

[46] “Intel® 64 and IA-32 Architectures Software Developer’s Manual,” Volume 3B, System Programming Guide, Part 2 , 2010

[47] P. Domingos and G. Hulten, ”Mining high-speed data streams,” In Int’l Conf on Knowledge Discovery and Data Mining , (SIGKDD), pages 71–80, Boston, MA, 2000. ACM Press.

[48] J. Han and M. Kamber, “Data Mining: Concepts and Techniques,” 2nd ed., San Francisco: Elsevier, 2006.

[49] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, ”Mining data streams: a review,” SIGMOD Record, 34(2): 18–26, 2005.

[50] http://2011.hpcs.ca/

[51] http://www.sigops.org/osr.html

[52] http://www.igi-global.com/bookstore/titledetails.aspx?TitleId=1123