TIME AWARE APPLICATIONS: PAST, PRESENT AND FUTURE

Calnex

John C. Eidson

WSTS-2014

www.calnexsol.com

A challenge from the past

Calnex

- When NTP and GPS made reasonably accurate time readily available, Barbara Liskov challenged the computer science community to "rethink" current algorithms and techniques. (Liskov, Barbara. "Practical uses of synchronized clocks in distributed systems." *Distributed Computing* 6, no. 4 (1993): 211-219}
- Sadly the response has been poor.
- There have been many others that noted the difficulties in controlling temporal properties especially in distributed systems. (e.g. Stankovic, John A. "Real-time computing system: The next generation." (1988)., Insup Lee, Susan B. Davidson, and Victor Fay-Wolfe. "Motivating time as a first class entity." (1987)., Lamport, Leslie. "Time, clocks, and the ordering of events in a distributed system." *Communications of the ACM* 21, no. 7 (1978): 558-565.}

So has explicit time been used in the past?

Of course- and extensively:

- Computer file and database systems (Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database system implementation*. Vol. 654. Upper Saddle River, NJ:: Prentice Hall, 2000.)
- Security protocols (B. Clifford Neuman, and Theodore Ts'o. "Kerberos: An authentication service for computer networks." *Communications Magazine, IEEE* 32, no. 9 (1994): 33-38.)
- Safety-critical systems (Kopetz, Hermann. *Real-time systems: design principles for distributed embedded applications*. Springer, 2011., Kopetz, Hermann, et al. "Distributed fault-tolerant real-time systems: The Mars approach." *Micro, IEEE* 9.1 (1989): 25-40.)

Time is heavily used today in safety-critical systems

Safety-critical systems- becoming even more important: power generations, transportation, many industrial processes.

More examples of current applications

Key learnings from safety-critical design*

Calnex

- Both time and value must be correct
- Bounded interaction with environment
- Composable subsystems=> time-triggered architecture
- Dependable => error containment
- Certifiable

*from Kopetz, Hermann. *Real-time systems: design principles for distributed embedded applications*. Springer, 2011.

Other traditional uses of time in systems

Calnex

Not all applications have safety-critical requirements, e.g. T&M typically does not, nor are time-triggered architectures appropriate for all applications.

Example: US Navy CASS test system: not safety-critical but still has significant timing issues

Hull Vibration Monitoring System

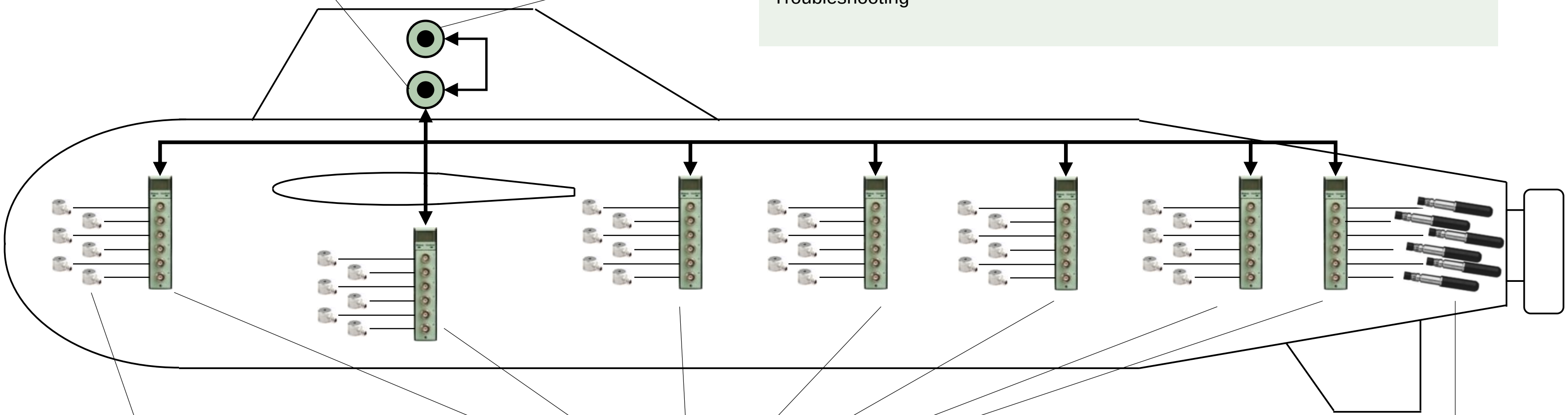
Calnex

Network Router

Precision Time Protocol Synchronization (PTP)
Power over Ethernet (PoE)
Data Transfer from LAN-XI Systems

Pulse DAQ-H Central Control System

Data Storage and Analysis
Interactive Customer User Interface
Automated and user initiated scans
Ability to listen to any signal
Warning/Alarms
Troubleshooting



COTS Accelerometers

Off the shelf product
Designed for Health Usage Monitoring Systems (HUMS)
High Frequency, High Temperature
EMI/Radiation Resistant

Distributed LAN-XI A/D Front-Ends

Dual 24bit A/D Technology (160dB Dynamic Range)
Extremely low Noise Floor
Single LAN cable operation (PoE, PTP, Data Transfer)
1000+ Channels in a single system

Embedded Hydrophones

Cavitation Noise
Integration of permanent hydrophones (Monitoring of Cavitation Noise)
Listening to signals

Slide courtesy of Bruel and Kjaer

Looking forward- Google’s Spanner

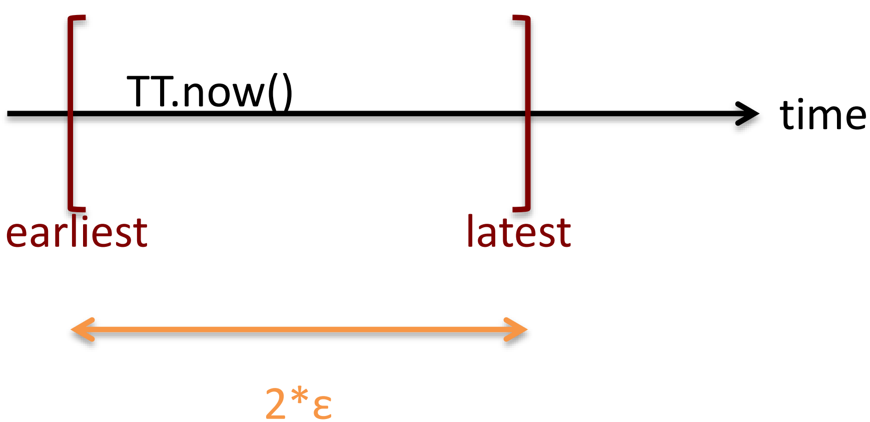
- Next several slides discuss Google’s Spanner database as an example of an innovative use of time*
- Spanner is a world-wide distributed database
- Achieves external consistency, i.e. users see same order as internal order
- Based on replacing reasoning about commit order with reasoning about timestamp order

*From: Wilson Hsieh, et.al. “Spanner: Google’s Globally-Distributed Database”, OSDI-2012

“TrueTime”

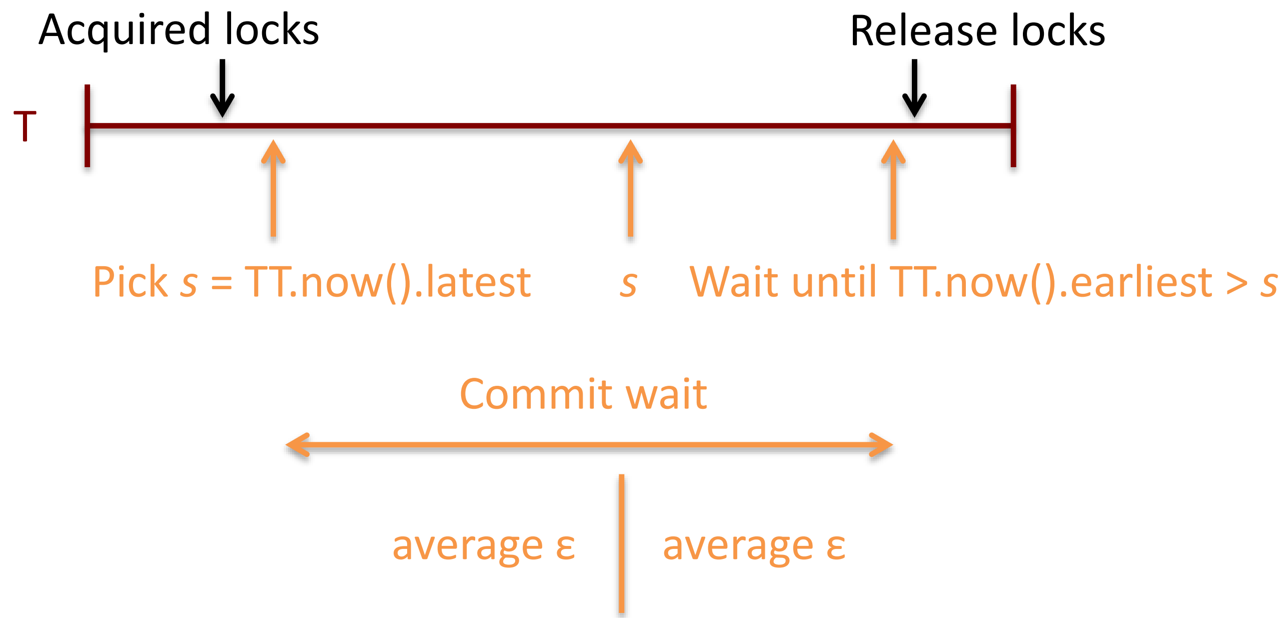
Calnex

- Global timescale with bounded uncertainty
- Same purpose as Kopetz’ sparse timescales
- Allows unambiguous ordering of events



Example of Timestamps Commit Reasoning

Calnex



Google's conclusion:

- Large scale no longer implies weaker semantics!
- Enabled by use of "True Time"

So why aren’t people rushing to use explicit time to develop applications?

Calnex

- Perhaps they don’t need to?, or
- Hard as real-time applications are (and they are hard) we can make do with the current techniques
- Still one wonders about future application requirements

Current level of "support" for use of explicit time

Calnex

Things you can't easily do today:

- Write a program that closes switch A at 10AM ±1μs (or even 100μs ±1μs after switch B is closed).
- Find out how long it takes to execute a piece of code.
- Order external events in a distributed system.
- Compose two components and understand the temporal semantics of the combination.

This is not because people have not tried to provide support!

To cite a few examples:

- Esterel, Lustre (SCADE), and Signal (synchronous languages)
- Real time Java, concurrent C, Ada …
- Giotto, TTP, TTE (time triggered)

This is not because people have not tried to provide support!

Calnex

Interesting European project*.

- Mapped from Simulink to SCADE to TTA
- At least one microprocessor target supported.
- Its fate: “Frankly speaking we do not see a very high interest from customers”

* Caspi, Paul, et al. "From Simulink to SCADE/Lustre to TTA: a layered approach for distributed embedded applications." ACM Sigplan Notices. Vol. 38. No. 7. ACM, 2003.

This is not because people have not tried to provide support!

Calnex

There are a number of development environments that have some temporal semantics:

- MathWorks ("Simulink Coder"), LabVIEW…
- Some industrial systems: e.g. Siemens Step7, TTTech toolsets

Maybe others will solve the problem IEEE CS 2022

“ Nine technical leaders of the IEEE Computer Society joined forces to write a technical report, entitled *IEEE CS 2022*, symbolically surveying 22 potential technologies that could change the landscape of computer science and industry by the year 2022.”

- 156 pages
- The word “time” is mentioned 61 times
  - Typically in a context like “time to market“
  - Only once as a critical technology “hard real-time <multicore> architectures with local memory and their programming “

What about the future?

Who knows? But there is an effort to find out.

Nascent effort beginning this Friday

- Based on a white-paper : “The Case for Cross Disciplinary Research on Time Aware Applications, Computers and Communication systems (TAACCS)”
  http://tf.nist.gov/seminars/WSTS/TAACCS/TheCaseforTAACCS.pdf
- Proposed collaborative effort between industry, academia, and government
- Target disciplines: Oscillators and clocks, time-transfer, time-aware networks, timing support for applications, development environments, and applications.
- Steering committee
- Initial meeting

Conclusions

- Things are tractable in the area of safety-critical systems
- We do not have general techniques for composing time-aware components
- Design and implementation with guaranteed temporal behavior remains elusive outside of a few very restricted environments
- There are lots of ideas floating around but some collaborative effort is needed.

Don’t forget …

Wednesday Night is Whisky Night

Calnex