The SimUTC Fault-Tolerant Distributed Systems Simulation Toolkit

Bettina Weiss

Günther Gridling

Ulrich Schmid

Klaus Schossmaier

Vienna University of Technology

Department of Automation

Oktober 26, 1999

Project SynUTC

http://www.auto.tuwien.ac.at/Projects/SynUTC

$\boxed{\text{Overview}}$

- Project SynUTC
- Simulation SimUTC
- User Interface EvalSys
- Example Plot
- Conclusion

$\boxed{\text{Project SynUTC}}$

Goal: Development of Clock Sync. Algorithm

- distributed, round-based
- achieves precision/accuracy of 1μs over LAN
- needs dedicated hardware (clock, network IFC card)

Problem

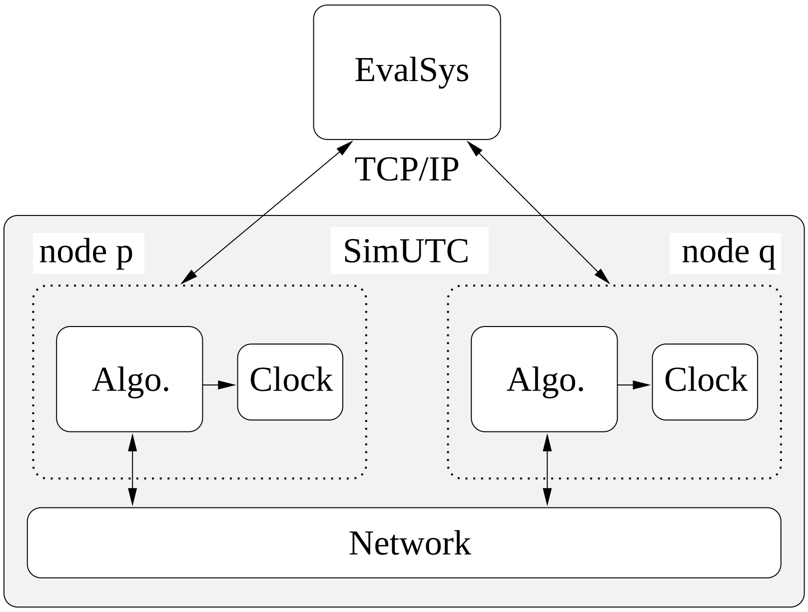
- algorithm was finished, hardware not
- need simulation to test the algorithm
- evaluate algorithm on hardware when available

Solution

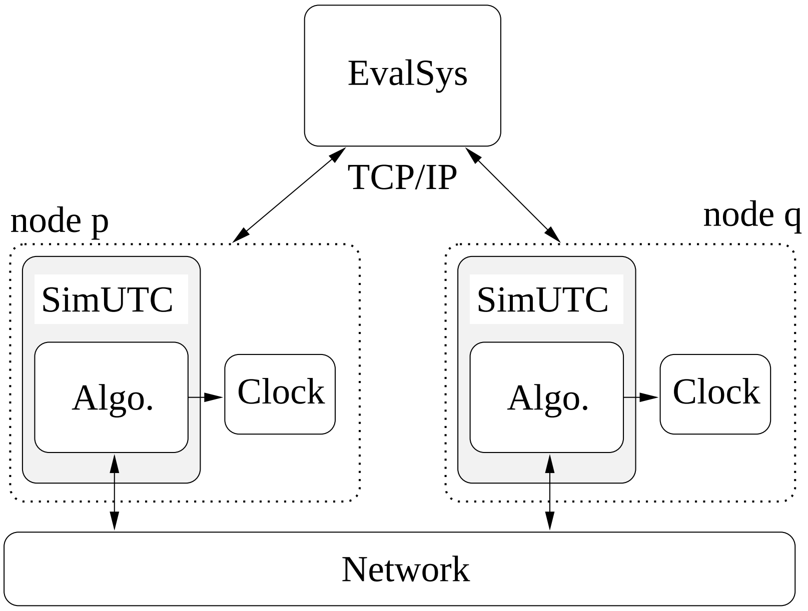
- build simulation that can migrate to hardware
- simulated HW can be replaced by real components
- the monitoring/fault injection parts remain
- plug in new algorithms

OVERVIEW SIMUTC

Simulation on Linux



Evaluation on Target Hardware



Components

Network

- transmission delays
- initial broadcast delay, broadcast operation delay
- message scrambling, lost messages

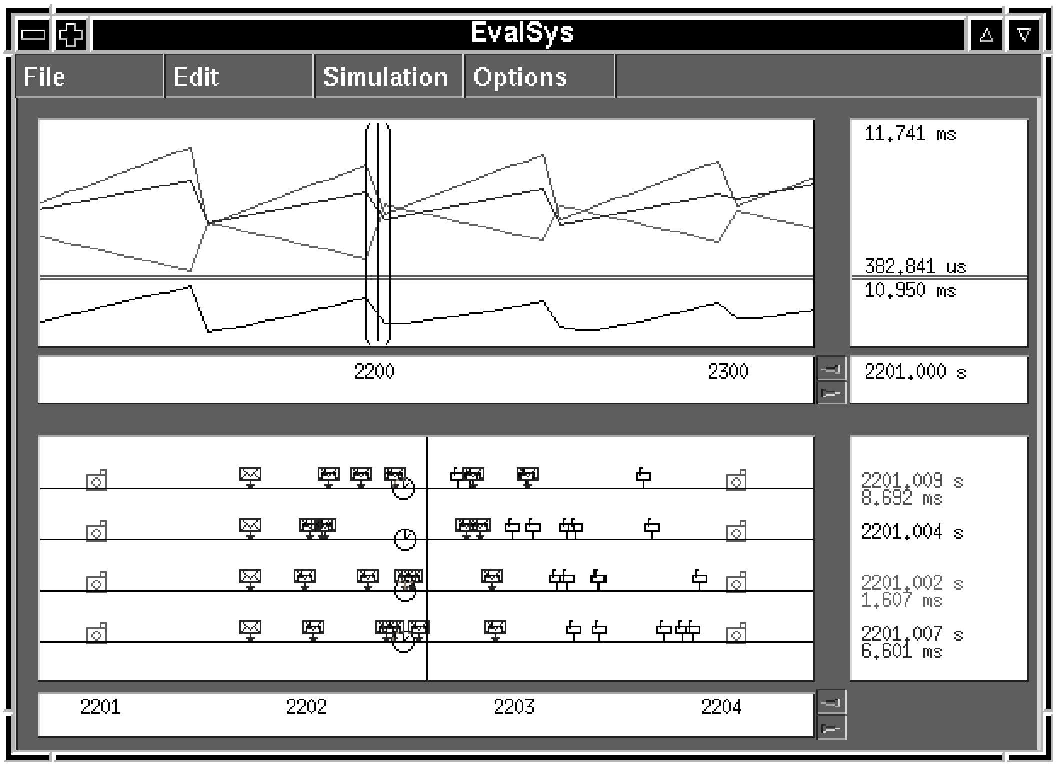
Clock

- clock drift
- read/write faults, clock stops

Algorithm

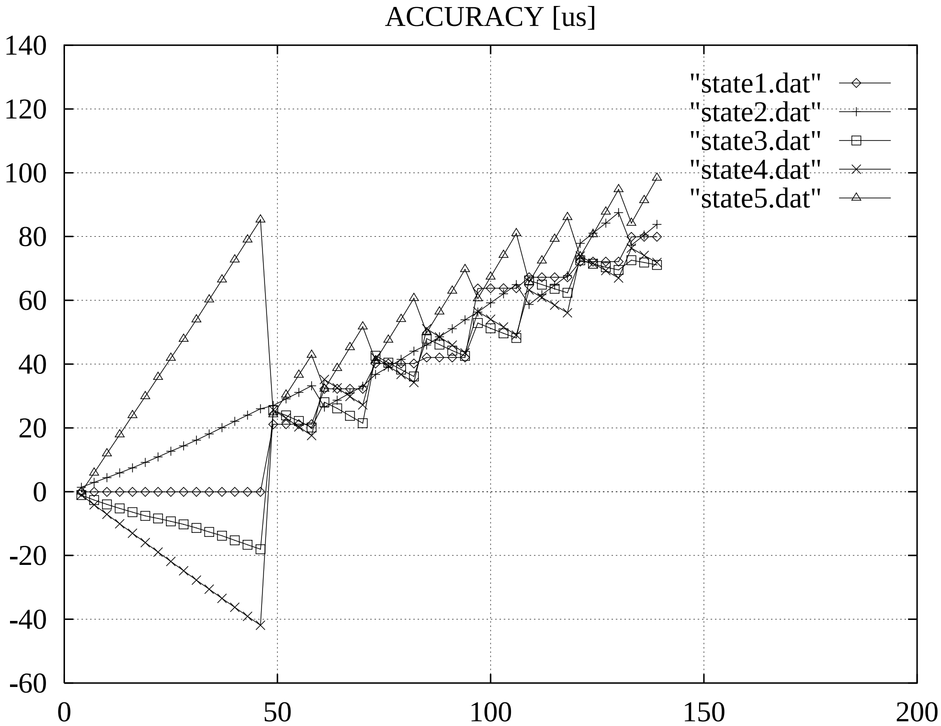
- establish timing of the round
- set number of nodes, number of faulty nodes

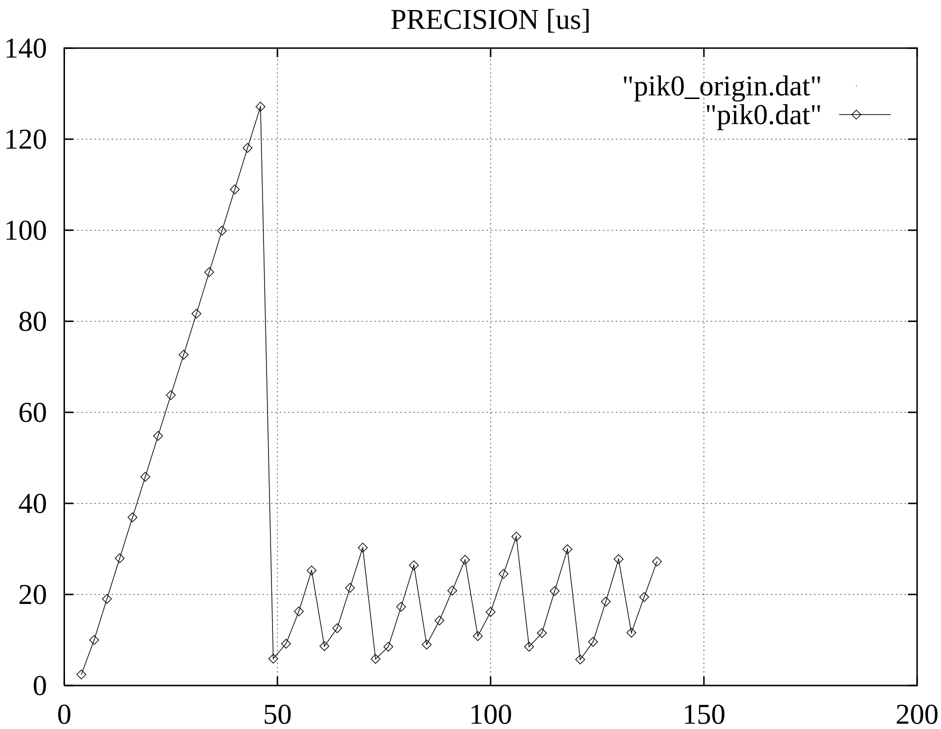
User Interface EvalSys



- data browsing
- configuration of simulation
- save simulation data for later processing
- gnuplot output of precision/accuracy

EXAMPLE


ACCURACY [us]
140
120
100
80
60
40
20
0
-20
-40
-60
0
50
100
150
200
"state1.dat"
"state2.dat"
"state3.dat"
"state4.dat"
"state5.dat"


PRECISION [us]
140
120
100
80
60
40
20
0
0
50
100
150
200
"pik0_origin.dat"
"pik0.dat"

Conclusions

SimUTC Features

- monitoring environment for clock sync. algorithms
- simulation as well as evaluation on hardware
- visualization separated from simulation
- graphical user interface
- new algorithms can be added

Future work

- experimental evaluation
- fault injection