Automated High-Accuracy Hybrid Measurement for Distributed Embedded Systems

Marcus Thoss

Distributed Operating Systems (DOPSY) Group
Fachhochschule Wiesbaden / University of Applied Sciences
Fachbereich Informatik
Kurt-Schumacher-Ring 18
D-65197 Wiesbaden / Germany

email: thoss@informatik.fh-wiesbaden.de
WWW: http://wwwvs.informatik.fh-wiesbaden.de

WISES 2005
May 20th 2005
Hamburg,Germany

Overview

Introduction

The challenge: timing measurements of distributed real-time systems

- Requirements:
  - high accuracy
  - common ("global") time base
  - using embedded systems: limited facilities
- Additional requirements for this study:
  - record single events
  - end-to-end delays of communication events
  - large number of measurement runs ⇒ unattended operation
- Pratical approach preferred, matching goals with minimum effort

Initial decisions derived:

- End-to-end delay ⇒ local clock timestamps not sufficient
- Dedicated clock source provides global time at observer
- Events are best generated locally
  - timestamp is determined by observer recording the event

  ⇒ hybrid approach
- Limited scope of "global" time as assumed here:
  - nodes within the experiment
  - time span covering the experiment
- Experiment conduction and analysis should be automated

Hybrid Approach

Combining local / centralised timestamp recording / processing

- Determining the occurrence of an event
  - event is determined locally
  - thus: close integration of event generation code and its context
  - events are exposed to an observer instance (device)
  - here: dedicated event signalling connections at hardware level

- Mapping of events to timestamps
  - exposed events are recorded by a central device
  - a high-accuracy clock source within the recording device is used to timestamp every event within a common time base ("global clock")
  - event context information (also exposed at event generation) is recorded with the timestamp for subsequent event identification/correlation

Hybrid Approach

Implementation Strategy

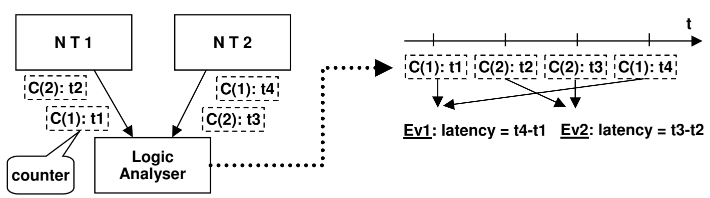
- Instrumentation of application code provides event signaling and counter value on PC parallel port

- Logic analyser records events with timestamps and counter value (context)

☞ *Customized* logic analyser recording state machine allows
  - long-term *and* high resolution time stamp acquisition
  - *single event* recording with multiple event sources
  - true end-to-end latency measurement (common time base)

Hybrid Approach

Correlation of send/receive event pairs based on counter values



Characteristics:

- 10 ns logic analyser resolution
- 17µs latency for event occurrence → logic analyser capture
  - mostly through I/O driver for parallel port access
  - 0.1 µs mean probable error

Modified Approach

- Motivation: greater number of nodes, more context data
  - capabilities of original implementation are exceeded

- Local clocks are used to capture timestamps
  - ⇒ common time base is initially lost

- For each measured node, establish a reference timestamp on the central timing device
  - reference timestamps are taken at beginning / end of measurement
  - procedure equals "original approach"

- Regain "global time" by mapping of local timestamps to logic analyser time
  - mapping is performed off-line after the measurement
  - timestamp data may remain distributed

Modified Approach

Mapping of local timestamps

- Assumed: merely linear clock drift within each node
- For reference timestamps $T_{LA1,2}$, local timestamps $T_{PC1,2}$ have been recorded

⇒ Local timestamps t_{pcn} can be mapped to "global" timestamps t_{LAn}:

$$t_{LAn} = (t_{PCn} - T_{PC1}) \cdot \frac{T_{LA2} - T_{LA1}}{T_{PC2} - T_{PC1}} + T_{LA1}$$

Approaches Compared

Limitations using local clocks

- Non-linear drift is not regarded
  ⇒ not suitable for variable processor clock speed and/or significant thermal changes affecting the clock speed
- Resolution and accuracy of the local clock must be sufficient

Benefits using local clocks

- Reduced overhead for taking timestamps
- No I/O access for event signalling: jitter is reduced
- Allows greater number of measurement nodes
- Context data can be more complex because it is kept locally

Automated Measurement System

Motivation for automated measurement, from study experiment:

- all permutations of parameter settings should be measured

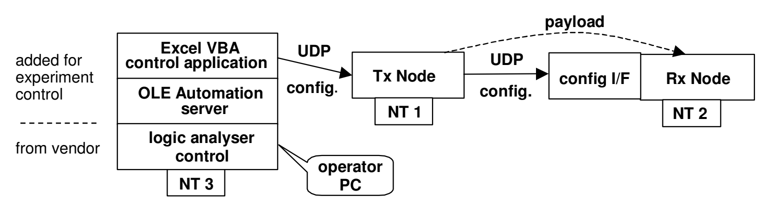
⇒ 400 measurements * 400 samples = 160,000 samples

⇒ experiment conduction / sample management and analysis likely to become tedious and error-prone

☞ Solution: Measurement control application on *operator PC*

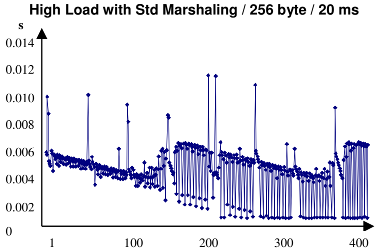
- experiment setup
- variation of experiment parameters
- control of experiment run/stop and data acquisition
- sample data management
- send/receive event correlation
- automated statistical analysis and diagram generation

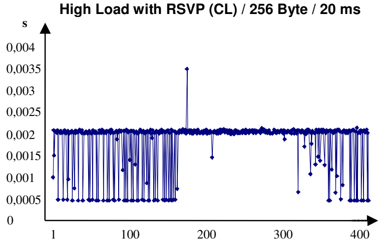
Automated Measurement System



- Operator PC runs control application as Excel VBA module
- NT 3 is connected to logic analyser, vendor provided control application is wrapped by custom OLE Automation server
- Experiment schedule UI for parameter preparation / execution
- Remote control of NT1 / NT2 through UDP communication
- Finally: automated sample import into Excel + analysis

Example Diagrams Showing Single Event Data





- Only possible using single event recording
- Singular exceptions visible (here: packets violating QoS)
- Appearance of artifacts, resulting from secondary effects

Summary

- Two aspects combined, forming a versatile measurement system
  - hybrid high-accuracy measurement
  - automated experiment conduction and analysis
- Single event recording preserving singular phenomena
- Automated operation: saving time, reducing errors
- Early judgements supported by analysis generation
- Usability and accuracy have been verified
  - accompanying application study
- Measurement system has been reused in several research projects
- Future extensions and enhancements planned