SPIDER: A Fault-Tolerant Bus Architecture

Lee Pike

Formal Methods Group
NASA Langley Research Center
lee.s.pike@nasa.gov

May 11, 2005
Motivation

- Safety-critical distributed x-by-wire applications are being deployed in inhospitable environments.
- Failure rates must be on the order of $10^{-9}$ per hour of operation.
Desiderata

- Integration
  - Off-the-shelf application integration
  - Off-the-shelf fault-tolerance
  - Eliminate redundancy
- Partitioning
  - Fault-partitioning
  - Modular certification
- Predictability
  - Hard real-time guarantees
  - A “virtual” TDMA bus

---

1John Rushby’s *A Comparison of Bus Architectures for Safety-Critical Embedded Systems*
Prominent Architectures

- TTTech’s Time-Triggered Architecture (TTA)
- Honeywell’s SAFEbus
- FlexRay (being developed by an automotive consortium)
- NASA Langley’s Scalable Processor-Independent Design for Enhanced Reliability (SPIDER)
“Time turns the improbable into the inevitable”
Collaborators

▶ Permanent Investigators
  ▶ Alfons Geser (formerly National Inst. of Aerospace)
  ▶ Jeffrey Maddalon (NASA)
  ▶ Mahyar Malekpour (NASA)
  ▶ Paul Miner (NASA)
  ▶ Radu Siminiceanu (National Inst. of Aerospace)
  ▶ Wilfredo Torres-Pomales (NASA)

▶ Industry Partners
  ▶ DSI, Inc.
  ▶ National Institute of Aerospace
SPIDER: A Fault-Tolerant Bus Architecture

Lee Pike
- Fault-tolerant time-reference and synchronization
- Diagnostic consensus and reconfiguration
- (Application-level) reintegration
- Communication with guaranteed consensus and latency
BIU/RMU Modes of Operation

- Self-Test Mode
- Initialization Mode
  - Initial Diagnosis
  - Initial Synchronization
  - Collective Diagnosis
- Preservation Mode
  - Clock Synchronization
  - Collective Diagnosis
  - PE Communication
- Reintegration Mode

Continuous on-line diagnosis...
A Hybrid Fault Model

- **Nonfaulty** The correct message is received at the scheduled time.
- **Benign** The message is detectably faulty by all receivers:
  - The message is received is outside the communication window.
  - The message is corrupted (or not present).
- **Symmetric** All receivers detect the same fault.
- **Asymmetric (Byzantine)** The messages received are arbitrary (in time and value).
- **Omissive Asymmetric** Each receiver determines the sender to be either nonfaulty or benign.
The Dynamic Maximum Fault Assumption

▶ For each BIU or RMU $i$, let $E_i$ be $i$’s eligibility set: the set of nodes $i$ believes to be nonfaulty.
▶ Let $N$ be the set of nonfaulty nodes.
▶ Let $B$ be the set of benign nodes.
▶ Let $A$ be the set of asymmetric nodes.

1. $2|N \cap E_i| > |E_i \setminus B|$ for all nodes $i$.
2. $|A \cap E_r| = 0$ for all RMUs $r$, or $|A \cap E_b| = 0$ for all BIUs $b$. 
Motivation

- Fault-injection testing cannot demonstrate $10^{-9}$ reliability
- Criticality warrants effort
- Complexity warrants effort
- Formal methods being integrated into certification standards
- Improved and structured design and understanding
Formal Methods Challenges

- Modeling faults
  - Variety of faults and locations
  - Nondeterminism in when they occur and duration
- Protocol/mode interaction and interdependence
- Protocols are distributed
- Protocols are real-time
- Varying degrees of synchrony
Formal Methods Tools for SPIDER

- Mechanical theorem-proving **PVS** (SRI)
- Model-checking and decision procedures
  - **SAL** (SRI)
  - **SMART** (William & Mary and National Institute of Aerospace)
- Interactive synthesis from Lisp-like language to a HDL
  **DRS** (Derivation Systems, Inc. and Indiana University)
Reintegration Overview

Allows a node that has suffered a transient fault to regain state consistent with the operational nodes. The node must regain:

- Clock synchronization
- Diagnostic data
- Dynamic scheduling data and other volatile state
- Developers: Wilfredo Torres-Pomales, Mahyar Malekpour, and Paul Miner (NASA)
- Formal Verification: Lee Pike (NASA)
The Frame Property

- $l$: number of faulty nodes not accused by the reintegrator
- $\pi$: maximum skew of nonfaulty nodes
- $P$: frame duration

$P > l\pi + 2\pi$
State Variables & Initialization

- **accs**: ARRAY of booleans, one for each monitored node
- **seen**: ARRAY of naturals, one for each monitored node
- **mode**: \{*prelim_diag*, *frame_synch*, *synch_capture*\}
- **clock**: $\mathbb{R}^\leq$
- **fs\_finish**: $\mathbb{R}^\leq$
- **pd\_finish**: $\mathbb{R}^\leq$

for each $i$, $accs[i] := false$;

$mode := prelim\_diag$;

for each $i$, $seen[i] := 0$;
Preliminary Diagnosis Mode

\[
pd_{\text{finish}} := clock + P + \pi;
\]
while \( clock < pd_{\text{finish}} \) do {
    for each \( i \), when echo(\( i \)) do {
        if \((seen[i] < 2 \text{ and not } accs[i])\) then 
            \( \text{seen}[i] := seen[i] + 1 \)
        else 
            \( \text{accs}[i] := true; \)
    }
    for each \( i \), if \( seen[i] = 0 \) then \( accs[i]; \)
    \( mode := frame\_synch; \)
for each $i$, $seen[i] := 0$;

$fs\_finish := clock$;

while $clock - fs\_finish < \pi$ do {
    for each $i$, when $echo(i)$ do {
        if ($seen[i] = 0$ and not $accs[i]$)
            then {
                $fs\_finish := clock$;
                $seen[i] := seen[i] + 1$;
            };
        else $accs[i] := true$;
    };

$mode := synch\_capture$;
for each $i$, $seen[i] := 0$;
while $seen\_cnt \leq trusted/2$ do {
    for each $i$, when $echo(i)$ do {
        if ($seen[i] = 0$ and not $accs[i]$)
            then $seen[i] := seen[i] + 1$;
    };
};
clock := 0;
Safety Properties

Theorem (No Operational Accusations)

For all operational nodes \( i \), \( accs[i] \) does not hold during the reintegration protocol.

Theorem (Synchronization Acquisition)

For all operational nodes \( i \), \(|clock - echo(i)| < \pi \) upon termination of the reintegration protocol.
Recent Successes

- A unified fault-tolerance protocol
- A fault-tolerant distributed system verification library
- Time-triggered schedule verification
- Case-study for research in model-checking, theorem-proving, and decision-procedures
Future Work

- Intrusion-tolerance
- OS and middleware
- Flight-testing
- Self-stabilization
Some Talks & Papers
http://www.cs.indiana.edu/~lepike/
Google: lee pike

SPIDER Homepage
http://shemesh.larc.nasa.gov/fm/spider/
Google: formal methods spider

NASA Langley Research Center Formal Methods Group
http://shemesh.larc.nasa.gov/fm/
Google: nasa formal methods