Different Paths to High Availability by Introducing Redundancy in a Distributed SCADA System

Morten Andersen

IT-Vest Thesis Examination – 27-06-2011 Master of IT – Software Development

Exam Question

"Med udgangspunkt i din rapport ønskes en præsentation af hovedideerne i din hovedopgave: motivation, problemstillinger samt hovedkonklusioner.

Herefter ønskes en detaljeret redegørelse for test opstillingerne og evalueringen (§ 7+8)."

Motivation

- Existing distributed
 SCADA system
- Soft real-time sensor monitoring
- Wind farm server: multiplexing forwarding observer
- Wind farm server: single point of failure



Motivation – Evolution



Problem Statement

"To **analyse** how the theoretical high availability tactics can be applied to the distributed wind farm SCADA system as an evolution of the existing system.

Based on this analysis, to **apply** several of these theories as architectural prototypes on a model of the distributed wind farm system.

And finally, to **evaluate** the suitability of the applied solutions."

Architectural QASes

- QAS1 (availability): 5 seconds deadline for sensor readings at failover time
- QAS2 (availability): 2 crashed wind farm servers
- QAS3 (performance): 1 second deadline for sensor readings at normal operation

Availability Tactics

Spare Wind farm server



Passive Redundancy

Active Redundancy

Reconstructable/Soft State End-to-End Principle

Passive Redundancy Prototype

- DSM based
 - One region per turbine
- Using Terracotta
 - DSM = network attached memory
 - Distributed objects
 - Consistency: Hybrid, entry consistency
- Transparency: Distribution of data is hidden



Active Redundancy Prototype

- Multicast based
 - Turbine nodes not part of multicast group
- Using Hazelcast
 - Explicit API library
 - JMS Topic alike interface
- Explicit communication



End-to-End Based Prototype

- Extend original solution
- "Redundancy in network routes"
 - Turbine nodes sees wind farm servers as individual clients – i.e. no group concept
 - "Multicast" (hand crafted) but not to entire group
- Re-computable / soft state

Evaluating the Prototypes

- Distributed system
- Hard to verify
 - Cost of hardware / setup
 - Non-deterministic

Evaluating the Prototypes

- Quantitative test (qualitative see app. F)
- Monte Carlo experiment
- Simulated network with random crashes
 - Repeatable sequence (known seed)
 - Within bounds (QAS2 maximum 2 crashed hosts)
- Amazon EC2 virtual server platform
 - Cost effective: price and time

Conclusions

Passive redundancy DSM based prototype

- Failing in fulfilling all QASes
- Significant network overhead
- Large code changes

Active redundancy multicast based prototype

- First impression: Solves all QASes
- Experiments revealed: QAS1 only fulfilled in 20% of tests
- Code changes feels natural

Conclusions

End-to-end based prototype

- Excels in fulfilling all QASes
- Complex state distributed responsibility
- Hard to maintain

Conclusions

Well defined test bench

- Solutions comparable
- Fact based decisions instead of based on feelings
- Generic test controller components

Amazon EC2

- Programmable / scriptable
- Cost effective
- Hard to debug (fact of multi-node system?)
- No control over node placement

Test Bench – EC2



Test Bench – EC2



EC2 – Network Transparency

- Network layout not transparent
- Risk of node co-residence
- Averse effects from other nodes



EC2 – Remedies

- Tests run on running instance
- IO Network measurements
 - Network latency fluctuations
 - No visible correlation: hops <-> latency
 - Fixed layout no live migration
- CPU measurements
 - Not performed applications not CPU bound
 - Indirect e.g. timing a known calculation

EC2 – Network Transparency

10.227.21.140 10.227.21.140 0.500 0.600 1.000 0.500 0.900 0.900 0.900 0.600 0.500 0.700 5 300 0.08 0 0.08 0 10.227.125.165 10.227.125.165 0.600 0.600 3.300 0.700 2.100 0 700 0 700 0.700 2.700 1 100 10.48.61.31 4 10.48.61.31 2 000 1.800 1 700 1.000 2.900 2.900 10.48.10.83 10.48.10.83 2 800 2 300 2 500 10.48.19.168 1 900 2 600 10.48.19.168 3 300 10.226.57.80 10.226.57.80 1.900 0.900 2.800 1.500 1.600 10.227.63.157 10.227.63.157 1.500 10.226.102.91 10.226.102.91 3,600 2,100 2,800 1.500 0.900 10.227.178.130 10.227.178.130 4.900 0.700 1.000 2.700 1.000 2.100 0.700 3,100 2,700 10.48.5.227 4 4 10.48.5.227 4.700 1,200 4.200 1,800 3,000 500 10.48.5.54 4 10.48.5.54 1.700 2,600 10.227.62.196 4 10.227.62.196 3.200 000 1.600 2.400 10.48.6.59 2 100 1 000 1 200 10.48.6.59 2,400 10.227.133.90 10.227.133.90 1 500 10.226.253.14 10.226.253.14 1 400 10.48.5.149 4 4 10.48.5.149 0.800 4.800 .700 0.600 10.48.237.132 10.48.237.132 1 000 2 000 1 500 0 50 2 800 4 10.227.61.136 4 4 10.227.61.136 2 100 2 500 36 83 65 4 65 4 54 Ö 33. 48.61. 02 18.61. 226.253 62. 78 63. 6 125. 10.227.21. .19 22 78. 33 25. 5 5 ģ 37 ģ ĝ 226 226 227 ġ 227 227 226 - 22 0.227 227. 227. ò ò ò 227 ò o Number of network hops between nodes Average network round trip time in ms. between nodes

Network Hops - 2011-06-06 11:02:24

Network Round Trip Time – 2011–06–05 13:33:36

EC2 – Network Transparency

10.227.21.140 500 0.500 0.600 1.000 0.500 0.900 0.500 0.900 0.900 0.600 0.500 0.700 5 300 0.800 0.600 1.000 0.700 0.000 10 227 21 140 .900 0.600 0.600 3.300 0.700 2.100 0.500 0.700 10.227.125.165 0 700 1 000 0.600 1.200 10.227.125.165 1.100 2.000 1.800 10.48.61.31 .400 0.700 2.700 0.800 1.700 1.000 2.900 2.900 10 48 61 31 10.48.10.83 200 2,800 2,300 1 000 0.600 0.800 1 300 0.900 2 500 3 200 10.48.10.83 1 000 1 900 2 600 3 300 102.700 10.48.19.168 0 900 2 400 0 500 1 100 1 500 10 48 19 168 10.226.57.80 1.900 0.900 0.700 2.800 0.900 1.500 1.600 10.226.57.80 10.227.63.157 0.600 1.500 0.000 2,700 101.400 10.227.63.157 10.226.102.91 4.500 500 1.500 1.400 0.000 2.100 2.800 101.500 100.500 1,200 0,900 10.226.102.91 10.227.178.130 4.900 1.800 0.700 1.000 2.700 1.000 2.100 0.700 3.100 2,700 1,000 1,700 10.227.178.130 10.48.5.227 4,700 2,600 1.600 1.200 4.200 000 1800 3.000 1.500 0.800 1.200 1.300 4.100 10.48.5.227 10.48.5.54 0.000 1.700 0.800 2,600 0,900 2,300 0.700 1.800 0.600 10.48.5.54 10.227.62.196 3.200 1.000 1.600 1.100 2.400 10.227.62.196 0.700 2.100 1.000 1.200 2.400 0.900 10.48.6.59 1.000 10.48.6.59 10.227.133.90 0.600 1.500 0.900 10.227.133.90 10.226.253.14 1.400 0.700 0.600 10.226.253.14 10.48.5.149 0.500 0.500 0.800 0.800 0.700 4.800 .700 0.600 0.800 10.48.5.149 10.48.237.132 1.000 2.000 1.500 0.500 2.800 0.800 0.600 0.600 10.48.237.132 10.227.61.136 2,100 2,500 10.227.61.136 65 4 18.61. 18.61 β. 63. σ 25. 10.227.21. S. 78. Ξġ. 25. 5 ģ ģ 226 227 226 0.227. 227. 227. 227 Average network round trip time in ms. between nodes

Network Round Trip Time - 2011-06-05 13:33:36

Average network round trip time in ms. between nodes

Network Round Trip Time - 2011-06-06 11:02:24

Test

- 100 test runs
 - 20 random length periods [20;40] seconds
 - Random server crash same state in max 5 periods
 - Per test length ~= 10-12 minutes
 - Total length ~= 24 hours
 - Average of 5-6 server crashes per test

Measurement

- Timestamp sensor readings on turbine node
- Record age when reading reaches client



Client Data Age - 2011-06-04 18:23:24

Age in ms. of data on one client - sampling every second

Prototype Evaluation – Terracotta

- 3. party library has big footprint
 - Startup time (+ 20 seconds)
 - Network traffic (factor 10, with larger peaks)

Prototype Evaluation – Terracotta



Age in ms. of data on one client – sampling every second

Age in ms. of data on one client - sampling every second

Prototype Evaluation – Hazelcast

- Low network overhead
- QAS problems in 80% of tests
 - 1) Startup problem (bug?)
 - 2) QAS1 occasionally exceeded

Prototype Evaluation – Hazelcast



Client Data Age - 2011-06-06 03:42:32

Client Data Age - 2011-06-05 14:37:03

Age in ms. of data on one client – sampling every second

Age in ms. of data on one client - sampling every second

Prototype Evaluation – End-to-End

- Succeeded on all QASes
- Segmented network not handled
- NIH syndrome?
- Complex to ensure correctness (fig. 29, p.63)
 - Distributed across nodes
 - Hard to maintain

Prototype Evaluation – End-to-End



Aggregated Client Data Age for 100 Runs – 2011–06–05 13:33:00

Age in ms. of data on one client – sampling every second

Client Data Age - 2011-06-04 23:24:19

Max age and mean in ms. of data for every test run

Prototype Evaluation – Hazelcast



Aggregated Client Data Age for 100 Runs – 2011–06–06 18:26:19

Aggregated Client Data Age for 100 Runs – 2011–06–06 18:25:51

Max age and mean in ms. of data for every test run

Max age and mean in ms. of data for every test run

Network Measurements

Wind Farm Server

Prototype	Kbps/sec in	Kbps/sec out
Passive (Terracotta)	6,000 (peak at 40,000)	10,000 (peak at 75,000)
Active (Hazelcast)	570	1,600
End-to-End	590	1,250

Turbine Node

Prototype	Kbps/sec in	Kbps/sec out
Passive (Terracotta)	150 (peak at 300)	600 (peak at 9,000)
Active (Hazelcast)	41	47
End-to-End	6	48

Critique of Test Method

- No statistical basic
 - Standard deviation
 - Confidence interval
 - Distribution
- Crash of management console during Hazelcast test sequence (test run #17)
- Missing network transparency
- Platform: Ubuntu vs. Windows

00:20:00

"The world is never going to be perfect, either on- or offline; so let's not set impossibly high standards for online"

Esther Dyson