WHITEPAPER

Strategies for Fault-Tolerant Systems Using Connext

AUTHOR

Paul Schmitt
Senior Field Application Engineer

ABSTRACT

Today's mission-critical safety-certified systems often require fault tolerance implemented with redundant field replaceable components or units (referred to as FRUs). The Data Distribution Service (DDS™) standard provides the building blocks for redundancy. This paper presents an overview of redundancy and three strategies for implementing redundant distributed controlling components using DDS.

FAULT TOLERANCE AND REDUNDANCY

Fault Tolerance is the ability of a system to continue operating, with little or no degradation in service, in the face of a hardware failure or software bug. Implemented by redundant components, fault tolerance is a "game of statistics" in that one needs to detect the failure, switch to a backup component to continue operating correctly, alarm the failed FRU, and allow replacement before a second failure. That is, minimizing the Mean Time to Repair (MTTR) is critical in reestablishing redundancy after an initial failure.

Ideally, redundancy guards against any single hardware or software failure. Redundancy relies on the probability that two failures within a short duration are highly improbable. With critical systems, controller redundancy alone is insufficient. A fully redundant system requires end-to-end redundancy, system isolation to avoid the effects of a cascading fault, and failure detection of any critical component. This means all system components, including the network, must be redundant and actively monitored to minimize the MTTR of any single FRU. Failing to detect and repair even a fault of a backup component allows the system to effectively operate in a non-redundant mode (referred to as 'simplex' operation) until an inevitable failure of a primary component occurs, leaving the system with no fallback position to continue operating correctly.

Figure 1 shows a basic fault-tolerant system. While all of the components are redundant, passive (e.g., the network), or receive-only (e.g., thrusters), components need only be monitored for failure and able to compensate for the loss of an element. In the example shown, while both port and starboard thrusters are used, any one port or starboard thruster should be able to adequately control the ship. The network is redundant.

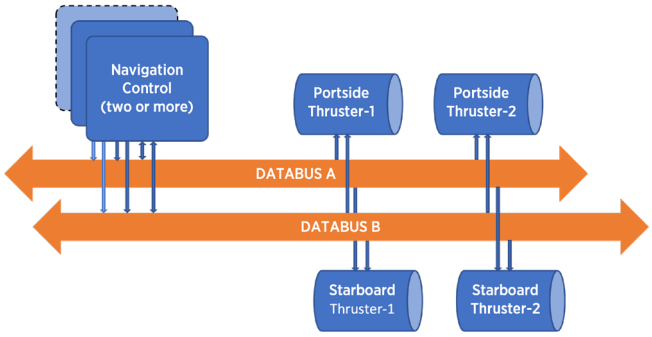

Figure 1. Basic Fault-tolerant System

Sending samples simultaneously on each network will work fine since DDS will mark and discard duplicate samples, allowing one copy of each sample to be sent to the application. Each thruster, along with both the A and B networks, requires some form of monitoring and alarm in the event of failure. Components that drive the system should have sufficient logic to determine which components are operating correctly in the case of a discrepancy.

REDUNDANCY ROLES AND STATES

A redundant logical component typically comprises two or more physical components defined by their role and state. The physical redundant components are often designated the roles of Primary and Secondary (or Backups), where the Primary controller outputs drive the system, and the Backup(s) stand ready to take over in the event of a failure. By definition, the Primary is in the Active State, indicating it is operational and driving the system. Each backup controller can be in any one of the following states: Hot-Standby, Warm-Standby, or Cold-Standby. In the case of a failure of the Primary, each standby state designation indicates a successively longer time to obtain the state necessary for the standby unit to take over as the Primary controller. Hot-Standby designates that the standby controller is immediately ready to go active. The switchover time for a Hot-Standby to go from Standby to Primary is the time to detect the Primary has failed. The method employed and procedure to obtain enough state to assume the Primary role depends on the specific system requirements.

There are also strategies that use no role or state. Here, each redundant controller actively drives the system where its outputs are used only if the other redundant components are in agreement.

SUMMARY OF FAULT TOLERANCE REQUIREMENTS

1. End-to-end
2. Timely detection of a fault on both the Primary and Backup units
3. Timely switchover to a backup component
4. Alarm and Repair of a failed component. This requires the redundant components to be field replaceable.

"Timely" is dependent upon system requirements.

REDUNDANCY STRATEGIES

There are multiple approaches to providing redundancy of the driving components. In the context of Figure 1, this is the Navigation Control. We'll look at three approaches, including:

1. Simple Duplex
2. Three-way Voting
3. Three-way Selected Sample

There is also a hardware fault-tolerant approach affectionately called "Pair-and-spare." Stratus Computer Inc. invented this design in the 1980s, but this approach only addressed hardware faults. Today, with far more reliable hardware, software faults must also be addressed.

SIMPLE DUPLEX

The Simple Duplex approach utilizes two off-the-shelf controllers, designated Primary and Secondary. DDS Exclusive Ownership Quality of Service (QoS) is used on the control topic with higher Ownership Strength assigned to the Primary.

Health monitoring could be implemented with an explicit periodic heartbeat topic, such as loss of Liveliness or missed Deadline QoS. This allows each controller to detect the loss of the other controller and raise an alarm for repair. DDS Exclusive Ownership QoS enables the Primary's data, with higher Ownership Strength, to be passed by DDS to the receiving controller application. Both controllers receive and process data in relative synchronization. If a failure occurs on the Primary, DDS will use the Secondary controller's data once the Deadline QoS is missed. Upon the original Primary returning to service, DDS will instantly resume utilization of the restored Primary's data based on the higher Exclusive Ownership Strength. This last step could result in an unstable Primary oscillating in and out of service. To prevent potential oscillation or a primary failure while still emitting data at the Primary ownership level (referred to as a zombie process), upon the Secondary taking over, the ownership strength could be raised to a value greater than the original Primary's ownership strength. That is, three ownership levels would be used: Primary (high), Secondary Standby (low), and Secondary Active (highest). At some point, the original Primary would need to be restored to the Active state to allow full redundancy to be restored.

This approach, while simple, assumes the failure mode is catastrophic (i.e., the process or controller completely fails). It does not cover more subtle issues of software bugs where both controllers send different results. Here, it is impossible to determine which controller is correct.

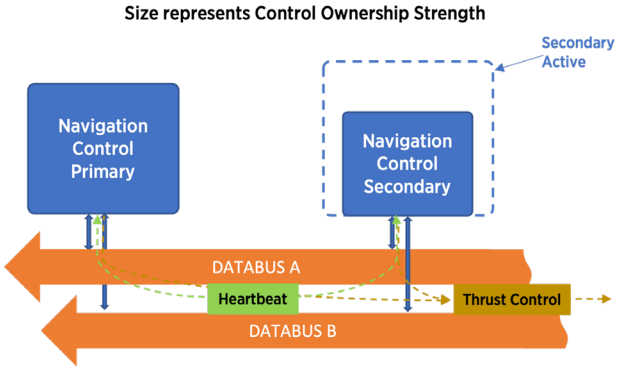

Figure 2. Simple Duplex

THREE-WAY VOTING

The Three-way voting approach utilizes three off-the-shelf controllers. The controllers will unanimously select primary, secondary, and tertiary roles by a majority vote.

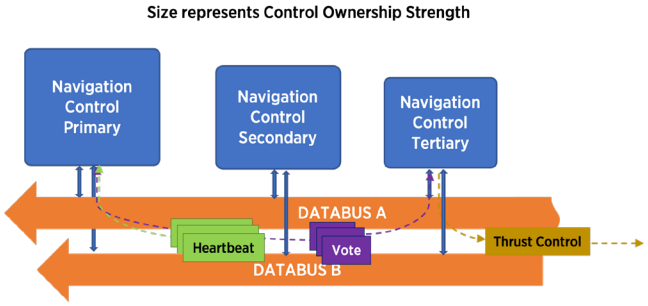

Figure 3. Three-way Voting

In this approach, a periodic Heartbeat topic could be generated to announce each controller and signify their state. Each controller would then assess these heartbeats and its own state for consistency, publishing a vote according to a common algorithm. Any dissenting vote is flagged as a failed unit and alarmed. The Primary will drive the system, with the Secondary and Tertiary standing ready to take over.

Technically, unless there is a double fault, the Tertiary would never drive the system and is used to validate a majority vote. DDS Exclusive Ownership QoS is used on the control topic with the highest Ownership Strength assigned to the Primary and the lowest Strength assigned to the Tertiary. Health monitoring would be implemented with an explicit heartbeat topic using DDS Liveliness QoS. This allows each controller to detect the loss of the other controller and raise an alarm for repair. DDS Exclusive Ownership QoS enables the controller's data — with the highest Ownership Strength — to be passed by DDS to the receiving controller application. All three controllers receive and process data in relative synchronization. Upon a loss or gain of a controller, the remaining controllers will re-elect the Primary, Secondary, and Tertiary, if applicable. If the Primary fails, DDS will use the Secondary controller's data once the Deadline QoS is missed.

It should be noted that the re-election process is independent of the deadline miss, where DDS uses the Secondary's data to drive the system. To prevent "Primary Role" Oscillation, upon a controller loss, controllers will promote existing controllers to the higher roles; that is, after the re-election process, there

will be at least a Primary and Secondary. New Controllers enter the lowest available role after the system is up (so-called late joiners). DDS Durability QoS on the Vote Topic will allow a new Controller to understand the system state and allow it to enter with the highest unfilled role.

Upon initial bootup, Three-way voting should allow a grace period to wait for all three controllers for full Three-way voting and redundant operation. The grace period timeout will enable the system to come up in either Simplex or Simple Duplex mode, as described above, allowing late joiners to come in later to reach full Three-way voted Redundancy.

Because this mode of operation is a superset of Simple Duplex, it is possible to include the techniques described above to guard against a zombied Primary failure mode. That is, upon promotion from the Secondary to Primary, the newly elected Primary could use a higher Primary Ownership Strength than the original Primary strength.

THREE-WAY SELECTED SAMPLE

The Three-way Selected Sample approach utilizes three off-the-shelf controllers. Unlike the previous two approaches, the Three-way Selected Sample has no defined roles, exclusive ownership, or ownership QoS. All controllers are peers and process the same input to produce the same output control commands. Commands are issued as Shared Ownership QoS. Heartbeat messages are used to detect and alarm the failure of a controller. With Selected Sample, the receiver application, in this case, the Thruster App, will receive all three sample instances, using a sample that matches with at least one other sample instance. Any dissenting sample instance will be flagged, and an alarm will be raised on the source of the sample. As described below, a Redundancy Layer should be used to separate out the Redundancy logic from the application business logic.

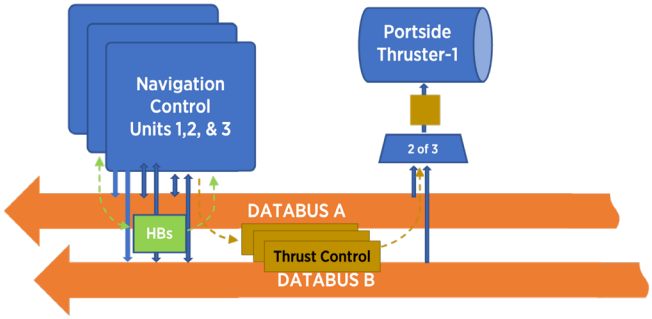

Figure 4. Three-way Selected Sample

This approach can impact performance as each sample passed to the application has to be compared in the Redundancy Layer software for a match with at least one other sample instance. Further, sample instances must be coherent in that a given sample instance sequence from all controllers must be presented as one collective set of samples (RTI Connext® has a Coherent Data Set mechanism for this). However, for highly safety critical systems, the advantage of guarding against software bugs offsets these complexities. Further software integrity can be achieved by using different sources to write the software for each controller to the same specification, making the likelihood of two or more controllers sending an erroneous sample at the same time, highly improbable.

REDUNDANCY LAYER

To isolate and abstract the applications in the system from a particular redundancy implementation, we introduce a Redundancy Layer (RL). The RL is responsible for the following activities (note activities are dependent upon the Redundancy Approach selected):

1. Active/Standby state selection
2. Failure detection and alarm
3. Switchover selection
4. Coherent Data matching and presentation
5. Network Monitor, Alarming and switching over

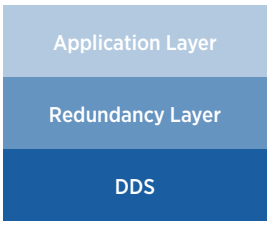

Figure 5. Redundancy Abstraction Layer

A single API is presented to the application, minimizing exposure of the redundancy implementation.

CONCLUSION

Modern mission-critical and life-dependent systems, such as defense systems, medical robotics and autonomous and semi-autonomous vehicles, are often widely distributed and require real-time performance to ensure data is delivered where, when, and how it's needed. DDS fits these applications well because it meets all of these demanding requirements, using fine-grained QoS and data-centric interfaces. DDS also supports important initiatives such as the Modular Open System Approach (MOSA), where the use of DDS is mandated by the U.S. Department of Defense (DoD).

Critical life-dependent systems often require redundancy to ensure they continue to operate in the face of failure. Using the highlighted DDS QoS building blocks, this paper presented three approaches in increasing redundant capability, depending on the criticality of the application.

APPENDIX

Table 1 highlights the Redundancy Strategies and attributes.

Redundancy Strategy	Cost	Complexity	Detect and overcome Hardware Fault	Detect and overcome Software Fault	Distributed	Switching Time (backup to primary)
Simple Duplex	Lowest	Lowest	Yes	Minimal, Total failure & Zombie Primary	Yes	Deadline based
Three-way voting	Low	Low*	Yes	Better, Basic software integrity checked by vote	Yes	Deadline based
Three-way Selected Sample	Low	Highest (multivendor)	Yes	Yes	Yes	N/A – Per sample selection (Impact performance)
Hardware "Pair & Spare"	High	High/custom hardware	Yes	No	No	Fastest (One CPU/Bus Clock cycle)

**Hardware "Pair & Spare" included for completeness*

Table 1. Redundancy Approaches and Key Attributes

Table 2 highlights the key DDS QoS mechanisms associated with the various approaches.

DDS Mechanism	Component Used	Description
Liveliness	Controller Components	Detects a redundant component/participant failure
Deadline	Controller Components	Time designated for a DDS reader to select the next strongest sample if a Primary component fails (misses its deadline)
Exclusive Ownership	Between Controller and Actuator	DDS QoS that enables a one controllers/writers sample to be selected among multiple controllers/writers of the same topic
Keyed Data	Controller Components	Enables redundent components to use liveliness to detect each controller in a 3-way voting system
Match on Publisher	Controller Components	Enables the system to determine how many controllers are active
Durability	Controller Components	Allows a newly or late joining controller in a system to obtain previously published state information allowing it to assume a consistent role within the system
Coherent Data Sets	Receiving Controller Component	For Three-way Selected Sample ensures three instances of the same sample sequence are collated prior to being processed for at least two samples being the same

Table 2. Key DDS QoS Mechanisms

FOR ADDITIONAL INFORMATION:

Please contact us at info@rti.com if you have questions or comments regarding this whitepaper.

ABOUT RTI

Real-Time Innovations (RTI) is the infrastructure software company for smart-world systems. Across industries, RTI Connext® is the leading software framework for intelligent distributed systems. RTI runs a smarter world.

RTI is the market leader in products compliant with the Data Distribution Service (DDS™) standard. RTI is privately held and headquartered in Silicon Valley with regional offices in Colorado, Spain, and Singapore.

RTI, Real-Time Innovations and the phrase "Your systems. Working as one," are registered trademarks or trademarks of Real-Time Innovations, Inc.
All other trademarks used in this document are the property of their respective owners. ©2024 RTI. All rights reserved. 50059 V1 0624

Your systems. Working as one.

CORPORATE HEADQUARTERS
232 E. Java Drive, Sunnyvale, CA 94089
Telephone: +1 (408) 990-7400
info@rti.com

rti.com
rti_software
rtisoftware
company/rti
rti.com/blog
connextpodcast