2013 NDIA GROUND VEHICLE SYSTEMS ENGINEERING and TECHNOLOGY SYMPOSIUM
Vehicle Electronics and Architecture (VEA) Mini-Symposium
August 21-22, 2013 – Troy, Michigan

SIMPLIFIED MIDDLEWARE TO INCREASE GROUND TACTICAL VEHICLE SYSTEM AVAILABILITY

Tri Nguyen
Software Architect
Northrop Grumman Corporation
Carson, CA

ABSTRACT

Northrop Grumman has developed Tactical Ground Vehicle High-Availability (HA) middleware conforming to open standards specified by the Service Availability Forum (SAF), a consortium of industry-leading communications and computing companies. The software hot-spare and standby capabilities realized by this technology operate across tightly and loosely coupled farms of processors, ensuring critical processes remain operational with zero or minimal interruption, as chosen by system architects.

High availability software delivers key benefits to the warfighter. Systems experience less downtime, helping to maintain continuity of tactical operations. Both hardware and software failures are managed, reducing the impact on system aborts and essential function failures and therefore reducing the number of computing elements required to meet system level availability SWAP-CC (Size, Weight, Power, and Cost, Cooling). The wrappers Northrop Grumman has created for open source and commercial implementations of the SAF middleware are specifically designed for use in tactical ground vehicles, both minimizing the impact on software development and also creating the possibility of integrating COTS software without modification of the COTS distribution. We shield application developers from most HA domain specific details such as startup, shutdown, failover and recovery policies, and so minimize HA training and impact to cost and schedule. Benchmark tests also show that the addition of HA services does not significantly increase system resources overhead even when the system is stressed in high message traffic scenarios.

INTRODUCTION

Tactical Ground Vehicles (TGVs) with advanced offensive and defensive capabilities enabled by high performance computers and digital networks can change the outcomes for our warfighters on the battlefield. Powered by software running on redundant clusters of processors and managed by middleware that maintains hot standby or fast restart of essential processes, these vehicles can continue to provide the processing needed to support telecommunications, situational awareness, and command control applications in the face of failures or combat damage. This paper describes how Northrop Grumman implemented the low cost technology we use to realize those benefits.

High availability automation built using this approach directly benefits warfighters. Functions including indirect vision driving, local situation awareness, communications, command and control, mission planning, target recognition, and fire control are all dependent on sophisticated software running on high performance computers, with only a severely degraded capability possible through reversion to manual backups.

We define *resilient computing systems* as composites of hardware and software able to leverage the remaining computing resources to maintain ongoing functions and missions in the event of damage or failures in hardware or software. Such systems permit crews to conduct combat and peacetime operations with very high availability of mission essential computing functions and low probability of system abort due to those computing functions.

AVAILABILITY

High availability is an attribute of systems that operate without interruption for long periods of time. Availability measures the uptime of a system, and is typically expressed as a percentage. A computer system with availability of 99.999% (commonly termed "five nines") is highly available, delivering service 99.999% of the time it is required. **Error! Reference source not found.** shows the total time a system may be unavailable in an entire calendar year and still meet availability of 90 percent and above. By the time availability requirements get into the 99 percent and above range, either very high reliability or automated mechanisms to restore lost functionality are required. In systems where such availability is required but damage is a factor, very high reliability is insufficient – a combination of redundant hardware and automated software process restoration mechanisms is the only way to maintain that level of availability.

Availability	Downtime per Year
90.0000%	37 days
99.0000%	3.7 days
99.9000%	9 hours
99.9900%	53 minutes
99.9990%	5 minutes
99.9999%	32 seconds

Table 1. Availability vs. Downtime

Suppose a system has been implemented as a composite of 1000 hardware and software components where all of those components must be operational for the system to function. If each component has 99.999 percent availability, the overall system will have 3.7 days down time per year, delivering availability of only 99 percent.

SYSTEM ARCHITECTURE

The key characteristic of a highly available (HA) computer system is its ability to provide uninterrupted service to its users in the event of hardware or software failure. No one approach achieves high service availability at reasonable cost; instead, practical HA systems implement a strategy of quickly replacing unavailable services using a uniform hardware infrastructure underneath recoverable software and data elements. Implementing that strategy typically combines these four mechanisms:

- Hardware, software, and data redundancy
- Automated recovery from failures when possible
- Minimized time to restore service
- Fault prediction and avoidance

Redundancy

An HA system must maintain two forms of redundancy. Spare hardware is required to ensure capacity remains available after failures, while redundant copies of software and (in some cases) data are required to ensure the information necessary to restart software and reload current state is available.

Until such time as literally self-healing hardware becomes available and practical, maintaining the level of hardware infrastructure necessary to support mandatory services in the face of damage and failure requires redundant hardware and the ability to migrate processing among hardware elements, routing around and avoiding failed ones. The degree to which architects must provision redundant servers, storage, networking, power supplies, and other components depends not only on the target availability, but also on the components' inherent reliability, vulnerability to damage, interconnection architecture, substitutability, repairability, and related factors. Uniformity of individual component characteristics and of the interconnections among them improves the substitutability of components, reducing the total online sparing required. Cost and volume may further be reduced if the requirement for spare hardware can be coalesced with that for hardware design margin.

Data Redundancy

Beyond what's required to provide spare processing and random access memory, hardware must be provisioned such that current copies of software and data are preserved over failures. Separating software images into a read-only partition, isolated from volatile data, can minimize the amount of read/write storage and traffic into and out of the supporting devices. Standard COTS (Commercial Off The Shelf) information technology practices including mirrored file systems, backup copies, or RAID (Redundant Array of Independent Disks) technology using data replication or distribution with parity over multiple disks address the software infrastructure requirements for data redundancy.

Software Redundancy

We provide service continuity by maintaining process execution in spite of failures of individual systems or components, and in spite of fault recovery, maintenance or system management actions. Process execution disruptions, regardless of cause, are dealt with automatically. The decision of what hardware resources support software continuation is made in a controlled manner using dynamic resource allocation.

We assume that data redundancy ensures information maintained in persistent storage is accessible to restarted processes. Because processes also maintain variable sin memory, software redundancy requires the state of the process – resources and data – be preserved and restored. There is a tradeoff between the currency of the restored state and the resourced dedicated to maintaining the state backup information; better currency requires more resources. For that reason, we provide a variety of software restart mechanisms for use depending on the needs of the individual process:

- *Concurrent redundancy* maintains a hot standby copy of the process instance. Any current service transaction continues uninterrupted after a fault of the live process once the fault is discovered.
- *Serial redundancy* is achieved by restarting the failed process as a duplicate instance, terminating the failed instance if necessary. The new service is available after the fault is detected and the duplicate process initialized. Service may be interrupted for a short period of time during initialization.

Concurrent redundancy imposes n overhead in terms of processor resource. It occupies more space, generates more heat, and consumes more power than is required for a non redundant system.

Serial redundancy is less costly in terms of size, weight, power, and cost, and with the exception of added latency before service is restored achieves a similar HA result as the concurrent approach.

IMPLEMENTATION

The Northrop Grumman High Availability Middleware is a set of C++ classes and libraries that simplify the work involved in implementing software that realize the concurrent and serial redundancy models. Experiments on the middleware by Northrop Grumman validated that the HA capabilities needed in TGVs can be implemented as a simplification of the far more comprehensive capabilities defined by the open Service Availability Forum (SAF) specification suite. Accordingly, we defined and wrote a middleware wrapper around an implementation of that specification that exposes only the functionality needed in the TGV application. This overall implementation approach – combining a TGV wrapper with proven, high-TRL COTS HA middleware – has several concrete benefits:

- Hiding unneeded SAF complexity reduces the cost and schedule for implementing HA in ground tactical vehicle applications, and increases the reliability of those implementations
- Wrapping the SAF implementation allows a choice of COTS implementation, because the fine details differentiating the open source implementation from a commercial one can be hidden by the wrapper
- Implementing the wrapper at the software middleware layer isolates application processes from underlying infrastructure. That isolation, in combination with a hardware computing platform that clusters multiple identical general purpose processors lets the C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) system avoid additional cost for HA-specific hardware
- The software-only HA solution permits design and integration-time tradeoffs between concurrent and serial redundancy, letting system designers optimize for SWAP far later in the design process than is possible with a hardware layer solution

What the SAF specification does not address, however, is the necessary data resilience capability. Rather than implement a proprietary solution, Northrop Grumman chose to integrate the open Data Distribution Service (DDS) defined by the Object Management Group. DDS defines a publish-subscribe data interchange capability with multiple ways to specify quality of service, including the ability to persist data independent of the publisher. The choice of a publish-subscribe data architecture both decouples software processes in general, simplifying system integration, and also provides a strong foundation for application integration. Northrop Grumman believes cross-application data integration via publish-subscribe methods would improve VICTORY (Vehicle Integration for C4ISR/EW Interoperability) significantly.

SERVICE AVAILABILITY FORUM (SA FORUM)

From its own description, "*The Service Availability Forum™ (SA Forum) is a consortium of industry-leading communications and computing companies working together to develop and publish high availability and related management software interface specifications*" [1]. SA Forum has published both an Application Interface Specification (AIS) and a Hardware Platform Interface (HPI). The AIS is what we exploit for the TGV application; the HPI provides hardware discovery, monitoring, and management capabilities not presently needed for TGV – particularly in light of VICTORY – and so is not further addressed here.

The AIS defines an abstraction layer between the application and the service availability middleware. Implementations of the AIS provide the HA framework used to support highly available applications. Figure 1 shows how the pieces fit together – a Northrop Grumman wrapper envelops the implementation of the AIS, providing a limited and specialized interface to TGV software subcomponents through a high availability process base class, and mediating the interactions of the AIS implementation with the platform via the Northrop Grumman TGV HA middleware itself.

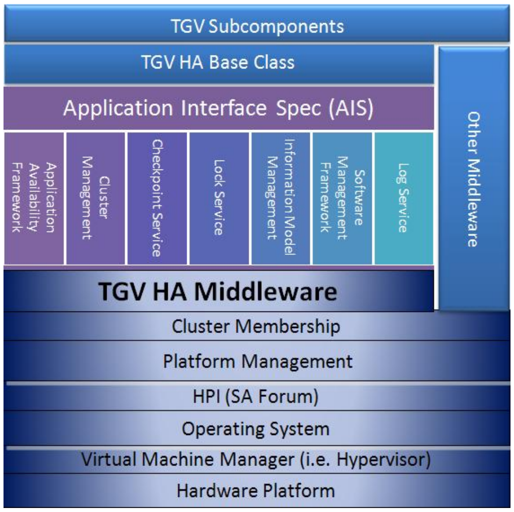

Figure 1. HA Architecture Tailored for Tactical Ground Vehicle

Detailed definitions of the non-TGV elements of Figure 1 can be found in the AIS; the portal to the specification is at [2].

DEVELOPMENT

Software development in the context of the TGV HA middleware has two parts. First, the overall system configuration is defined – once – for the processes and processors managed by the HA middleware. Each managed process then is written incorporating calls into the simplified API (application programming interface) of the TGV HA middleware process base class.

HA configuration defines the context in which HA processes run. Developers invoke the HA middleware to do the following:

- Identify the hardware environment, which consists of processing entities such as blades, nodes, and/or clusters. Asymmetric combinations are allowed
- Identify the logical software processes and services (Service Units and Service Groups in SAF terminology) within the hardware environment
- Identify the strategy used to recover processes and services after node failures. Unique strategies can be employed for each process and service

Configuration options typical for multiprocessor environments are visible through the TGV HA middleware, including the ability to bind processes and services to one or more unique hardware elements, to define where processes and services start under normal or other conditions, and to define which processes and services are inessential in a constrained environment. Resource allocation is dynamic, and accounts for actual resource availability.

Northrop Grumman chose to hide the SA Forum HA middleware API from TGV applications by creating an abstract HA base class that handles the details of the necessary interactions with the SA Forum HA middleware. Each TGV process type required for the applications becomes a class derived from the base type, inheriting its methods and adding new ones as appropriate. Initialization within processes involves little more than registering a few callback functions, as illustrated in this example:

```
// Register Callback to HA state change
m_HaState = HA_INITIAL;
c_HaManager *HM = c_HaManager::CreateInstance();
HM->RegisterHaCallback(c_CopAgent::_HaStateChangeCB);

// Register Callback for Event Pub/Sub
c_EventManager *EM = c_EventManager::CreateInstance();
EM->RegisterEventCallback(c_CopAgent::_EventSubCB);
```

The states presented through the state change callback are those defined by SA Forum, maintaining consistency with their middleware implementation. The publish-subscribe events are new entities created by Northrop Grumman to facilitate the integration of DDS with HA.

Checkpoints are the SA Forum mechanism to control the transfer of state information from a previously live process or service instance to the recovered instance. We use the standard, unmodified SA Forum API to access checkpoints; the DDS publish-subscribe mechanisms integrated with checkpoints in a simple, direct manner.

PERFORMANCE

OpenClovis Solutions, which provides software and support based on the SAFplus open source code base, has measured the performance impacts of the SA Forum standards using a variety of hardware configurations. The following performance data are summarized from their report. The measurements date from 2007, however, so in this paper we've attempted to correlate the hardware tested to what it might correspond to today (mid-2013).

- *Low-end*: The tested 4 node cluster of Pentium 4 and Celeron processors running several versions of Linux at speeds from 1.3 to 2.53 GHz might correlate with a federation of embedded controllers. Times to fail over an application where a hot standby was running, including time to detect the failure, ranged from 27 to 112 milliseconds depending on processor loading.
- *High-end*: The tested 10 node cluster of quad-core Xeon processors at 2 GHz corresponds loosely to a cluster of Sandy Bridge i7 quad-core single board computers. The test looked at failover of an application from one node to another, so it can be argued that the size of the cluster wasn't highly relevant to this test. Failover times, including time to detect the failure, ranged from 21 to 118 ms depending on processor loading.

Independent of any attempt to rationalize the similarity in the two sets of measurements, the times to detect failure and recover functionality are well within the timeframe for which interactive processes would appear to have been continuously available. Real time processes with hard timing constraints would require more careful analysis, engineering, and provisioning.

APPLICATION AND SERVICE RESILIENCE – VICTORY

IBM classifies application resilience into five categories [3].

1. *No application recovery*
2. *Automatic application restart and manual repositioning within applications*
3. *Automatic application restart and semi-automatic recovery*
4. *Automatic application restart and automatic recovery to last transaction boundary*
5. *Full application resilience with automatic restart and transparent failover*

The Northrop Grumman HA middleware provides simplified integration of highly available processing support at levels four and five of the IBM categorization. Taken in concert with our integration of DDS support to improve the ability of systems integrators to tie disparate applications together on the same computing and display platform, this technology operating at the highest levels of the IBM taxonomy could enhance the value of the emerging U.S. Army VICTORY standard. Because so many of the services standardized by VICTORY may be mission critical (e.g., Time Sync, Position, Orientation, Direction, Audio/Video, C4, EW services), integration of HA technology at low cost and low complexity into the VICTORY standard would greatly benefit future warfighters.

CONCLUSION

High availability, resilient applications and services can be achieved simply and at low added cost using open standards when implemented by highly mature COTS specifications and software. Integration of that technology, the open DDS standard, and VICTORY could be the answer to the common operating environment that for so long has been the Holy Grail of vehicle integration, while at the same time improving C4ISR system availability and lowering system integration costs.

REFERENCES

[1] Service Availability Forum, "The Service Availability Forum and Open Specification Solutions", February 2009

(http://www.saforum.org/HOA/assn16627/images/The_Service_Availability_Solution_FINAL_Whitepaper.pdf)

[2] Service Availability Forum, “Application Interface Specification”, http://www.associationvoice.com/Service-Availability-Forum:-Application-Interface-Specification~217404~16627.htm

[3] IBM, *IBM i 7.1 Information Center, Availability, High availability, High availability overview, Components of high availability, Application resilience,* http://pic.dhe.ibm.com/infocenter/iseries/v7r1m0/index.jsp?topic=%2Frzarj%2Frzarjcompappres.htm