Real-time Simulation and Model-Based-Systems-Engineering using Commercial-Off-The-Shelf (COTS) Computer Equipment

Timothy Magisos
Applied Dynamics International, Ann Arbor, MI, 48108

This paper presents the requirements development (including requirement creep), selection, and integration of a small commercial off the shelf (COTS) computer into existing simulation systems that are based on the Applied Dynamics ADvantage Framework. A comparison of the increased processing and modeling capability will be presented. Finally, changes in the existing process are shown with a discussion of modeling tradeoffs.

I. Introduction

At Applied Dynamics International (ADI) we run into 3 types of real-time simulation systems, rapid prototyping systems, simulation systems, and verification/test systems. These systems are being used in programs that can last in excess of 20 years. The length of these programs guarantee that technology will improve. In many cases the complexity of the simulation will increase as new features or capabilities are added stretching the capability of the processors in these systems. Most of these systems will be incorporated into a rack along with input/output devices (IO). The IO may be PCI, PXI, or VME based, or even custom, with the existing compute engine as the controller. In many cases the racks are well populated and space is limited.

We are all aware of Moore's Law, which has relatively accurately predicted a doubling of processing power every two years. For long-life programs, the opportunity to leverage these gains is immense. Significant processing power gains can be available for even a system in service for a short number of years. COTS computer equipment can potentially provide a good solution. In addition, more and more form factors are becoming available off the shelf, making choosing one that closely fits the situation easier. The question to be answered is: Can they be integrated in a timely, cost effective way to provide additional computational power and capabilities in an existing system?

Integration of a new COTS computation engine into an existing system poses some challenges. Aside from the form factor mentioned above, additional factors are the real-time operating system, and the modeling, simulation, and testing tools already in use. Existing systems always come with a process, a process that most will want to maintain, leading to the most challenging requirement, "Make it work the same!"

This paper will present the requirements development (including requirement creep), selection, and integration of a small COTS computer into existing simulation systems that are based on the ADI ADvantage Framework. A comparison of the increased processing and modeling capability will be presented. Finally, changes in the existing process will be shown with a discussion of modeling tradeoffs.

The ADvantage Framework is a Model Based Systems Engineering platform providing a feature-rich environment for supporting system lifecycle through development, integration, and test. The open architecture framework allows you to leverage best-in-class COTS and open source technologies in a common project based environment. Integration of a small form factor COTS compute engine needs to allow access to all of the capabilities of the ADvantage Framework.

The ADvantage software tools are ADvantageDE, ADvantageVI, SIMplotter, ADvantageDB and ADvantageFC. ADvantageDE combines hardware with open source software technology to bring together models and real hardware to create a coherent, reconfigurable Advantage project. ADvantageVI provides an easy-to-use interactive test environment which allows you to connect ADvantage based systems and test to operational limits and beyond. SIMplotter provides one resource throughout the product life cycle to collate, analyze and manage data; one common configurable resource to manage data representation from development platforms such as The MathWorks MATLAB/Simulink, though full-fledged integration test platforms such as ADI' ADvantage Framework. ADvantageDB provides an open-architecture, multi-protocol tool to define, manage, analyze, and share system network configurations. ADvantageDB lets you easily apply configurations to real-time test facilities to drive your integration and verification processes. ADvantageFC provides the capability to insert electrical faults. The hardware used in the ADvantage Framework includes real-time hardware based on both Intel based processors using the QNX operating system and Motorola Power PC based Single Board Computers (SBCs) running ADI's real-time kernel, RT-Exec.

II. Real Time Simulation Systems Assets

In "The Expanded Reach of Simulation Based Aircraft System Verification and its Software Capability Requirements", ADI's president Scott James and head of applications Dr. Clare Savaglio reviewed the historical trend of model based systems verification for aircraft, reviewed the traditional methodologies and facilities that have become the industry norm. These systems have been part of the process for many years. Real-time simulation systems have been in use for generations in a variety of industries. For many programs, including aircraft, these systems have to be maintained for Product Lifecycle Management (PLM).

The list of systems with life cycles in the 20-30 year range is long. One only has to look in the news to see aircraft such as the B52 that continue to perform long past expectations. As pointed out in the above reference most of these programs use verification systems and facilities that involve real-time simulation systems. As the role of those platforms expand, their capabilities expand, and the platforms themselves are modernized, the role of the verification system, and the included real-time simulation systems, needs to keep pace.

Starting in the early 1990s simulation systems started to use more model based digital real-time simulation systems for verification and testing. CPU processing power, input/output devices, and development tools to support those systems were expensive. In addition, these simulation assets needed to be available for use over the life of the program, a lifecycle, depending on the success of the system, could last 20+ years.

Product lifecycles lasting more than a few years will experience requirement creep. Expanded roles, increased capabilities, and integration with other systems will require that the verification systems, and hence, the simulation systems involved, will need to be updated and upgraded periodically.

III. The ADvantage Framework

The ADvantage Framework, from Applied Dynamics International, is a Model Based Systems Engineering (MSBE) platform providing a feature-rich environment for supporting system lifecycle through development, integration, and test. The open architecture framework allows you to leverage best-in-class COTS and open source technologies in a common project based environment. Real-time simulation systems utilizing the ADvantage Framework are an important part of many test and verification systems and have been for many years. The ADvantage Framework consists of ADvantageDE, ADvantageVI, SIMplotter, ADvantageFC, ADvantageDB, and ADvantage run-time services. For the purposes of this paper ADvantageDE, ADvantageVI, SIMplotter, and Run-Time Services will be used.

A. ADvantageDE

ADvantageDE, shown in Fig. 1, is a project based environment that brings together software models and real hardware to create an ADvantage project. An ADvantage project can contain one or more model assemblies such as Simulink, C, C++, Fortran, and SystemBuild. ADvantageDE is also where graphical user interface (GUI) assemblies, such as Altia, LabVIEW, and ADI's ADEPT are integrated into the ADvantage project. ADvantageDE allows selection of various types of real-time simulation targets from a range of COTS PCI, PXI, and VME computer equipment to be tied to the modeling and GUI assemblies. Input/output devices are tied to the models and GUI assemblies using a software application layer called a logical device. Input/output hardware devices are tied to software model input/output ports through the use of input/output ports in the logical devices. Use of the logical device concept keeps model assemblies hardware independent.

Distributed simulations, simulations using multiple targets, can be configured in ADvantageDE using the concept of framework projects. A framework project can contain one or more ADvantage projects. Model ports from one ADvantage project are connected to another ADvantage project using dedicated networks. These networks can be based on either Ethernet or SCRAMNet technology.

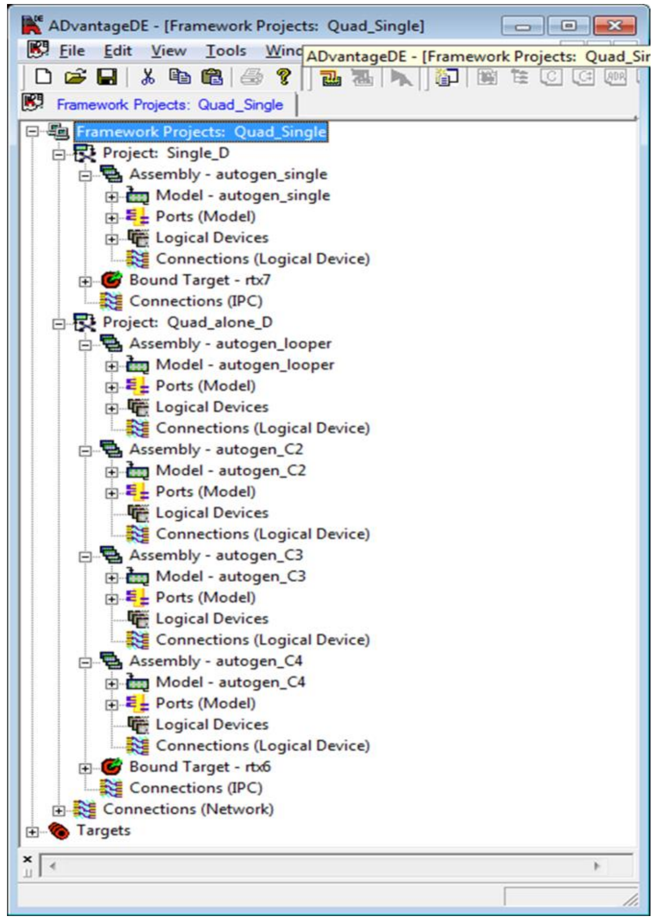

Figure 1. ADvantageDE a project based development environment.

B. ADvantageVI

ADvantageVI, shown in Fig. 2, provides a feature rich, easy-to-use interactive test environment. The ADvantage Framework operates with a host and target architecture. The ADvantageVI application runs on a PC connected to the target across standard TCP/IP Ethernet. ADvantageVI provides visibility into the target, its processors, I/O, and models. In addition, runtime statistics can be collected to quantify performance.

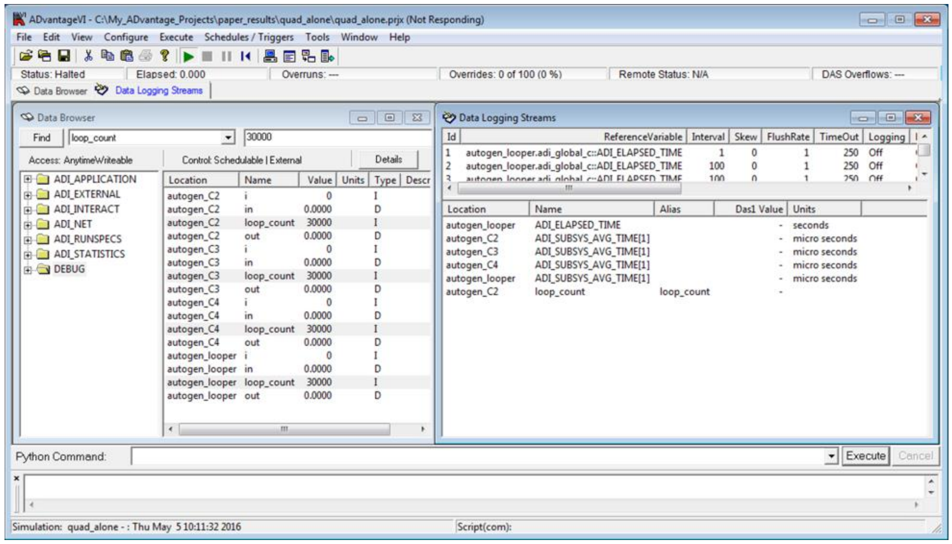

Figure 2. ADvantageVI an interactive test environment.

In a distributed, multimode project that configures multiple targets as a single synchronized system ADvantageVI connects to each target, seamlessly manages synchronization, and allows the system to be operated as a small real-time simulator.

Test management in ADvantageVI is accomplished via command line input and Python scripting. The scripting environment supports high level automation of data acquisition, test analysis, and report generation.

ADvantageVI supports the concept of data dictionaries as the specification for the interface into model, user application, or databus. Each model assembly includes a data dictionary, automatically generated from the model with inputs, outputs, signals, parameters, buses, etc.

C. SIMplotter

SIMplotter, shown in Fig. 3, provides a visual resource throughout product life cycle to collate, analyze, and manage data. One common source to manage data presentation from development platforms. Simplotter creates high quality plots intuitively using drag and drop interactively using Python scripting. SIMplotter can connect to live streaming data from a single, multiple, or distributed sources using TCP/IP ports. Data can be loaded from existing files for review, analysis, post-processing, and report generation. Overplotting of historical data with live streaming data provides easy comparison of live and truth data.

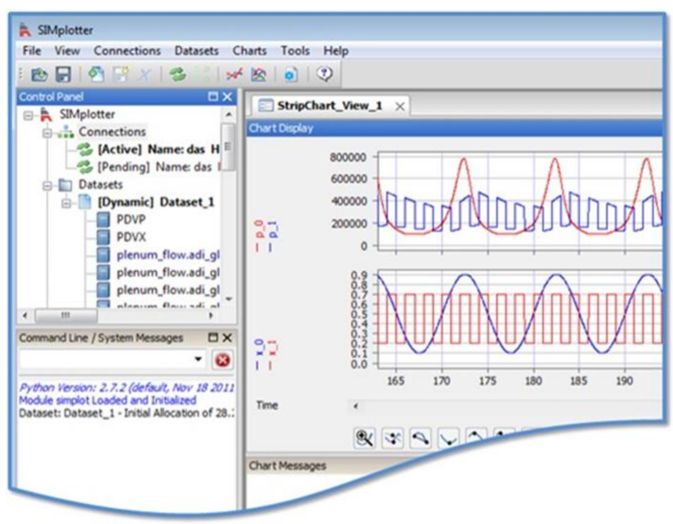

Figure 3. SIMplotter a tool to collate, analyze, and manage data.

D. ADvantage Run-time Services

The ADvantage Framework includes a set of services (i.e. libraries providing software capability) executed on the target computer. These run-time services enable the ADvantage tools to interface with the target and provide a range of capability for system interaction, visualization, data collection, real-time and non-real-time scripting. The run-time services are installed on the target hardware and interface to the general purpose or real-time operating systems through the POSIX layer.

The run-time services include the ability to "put" and "get" values in memory for inspection and control, collect real-time data, time stamped data acquisition of memory values, event and time based real-time scripting, loading and execution of model assemblies, control of I/O interfaces, and scheduling synchronous and asynchronous execution of these services.

IV. Program Requirement Creep

Programs that have been around for a while will experience program requirement creep. Program requirement creep is caused by the expansion of the role that the system is required to respond to, or creep in the requirements that the verification system is to test and verify. In either case the creep at the simulation level means increased model fidelity. For example, in the initial testing system the simulation system needed to model the flight dynamics of an aircraft. Later in the program additional elements in the airframe need to be modeled, such as communication buses. Dependent on the overhead in the existing simulation system, this can require additional processing power which may necessitate replacing, or upgrading, the simulation system.

Simulation assets, such as verification systems, represent a significant program investment. Especially when considering the training of development, test, and maintenance personnel. In many cases some level of recertification of the process and equipment is required. From this perspective it makes sense to upgrade the existing system, to keep what works, and minimize the amount of recertification.

V. Simulation Asset Upgrade

This paper presents the effort to create an upgrade path for existing simulation systems that require additional processing power to accommodate larger simulation models. The goal was to find a COTS computer with a small foot print and minimal environmental requirements that could be added to a simulation system based on the ADvantage Framework.

A. Existing Simulation Systems

Existing simulation targets that are ADvantage based consist of 3 types; rtX, RTS, and rtX-V. In some cases more than one target may be used in a distributed simulation. The rtX, RTS, and rtX-V are high-performance, standards based, open-architecture real-time system for hardware-in-the-loop simulation, controller prototyping, and test applications.

The rtX features the latest Intel multi-core processors for primary computation power. A wide range of PCI, PXI, PCIe, and other off-the-shelf interface cards can be used to create a system. The rtX utilizes the POSIX-compliant, high-performance, QNX RTOS as the base operating system.

The RTS is a VME based solution that features PowerPC VME single board computers (SBCs) for the primary computational power. A wide range of VME, PMC, and IP off-the-shelf IO interface cards are used to create a system. The RTS utilizes the RT-Exec small footprint real-time kernel as the base operating system.

The rtX-V is a VME based solution that features Intel multi-core VME single board computers (SBCs) for the primary computational power. A wide range of VME, PMC, and IP off-the-shelf IO interface cards may be used. The rtX-V utilizes the POSIX-compliant, high-performance, QNX RTOS as the base operating system.

The most common use of these targets is in a rack with the program's required IO and, in many cases, customer equipment. Space in the racks in existing systems is quite often limited.

It was decided to use an Intel COTS based system. Searching various venders online, the Jetway JBC375F3AW-2930-B Barebone kit from MITXPC was chosen. Fig. 3 shows this COTS computer. This computer measured 7.09" W x 5.59" D x 1.89"H. This system provided an Intel Celeron N2930 Quad Core

Figure 4. JBC375F3AW quad core barebone kit from MITXPC.

processor with 1.84 GHz processing speed with a 60 GB hard drive and 2 GB of RAM. This would add a substantial increase of processing power to older systems with Intel single core processors or PowerPC SBC processors. The small footprint makes it suitable for installing in even the most crowded rack.

B. Small COTS Quad Core

The JBC375 was installed with QNX, the ADvantage run-time services and configuration files were created. Testing was done to confirm correct operation of simulation models on all 4 cores. The best case processing power was quantified by creating a simple for loop C code model assembly. Fig. 5 shows the ADvantageDE project. There are 4 model assemblies, one for each core. The number of for loops was parameterized. The simulation was started and the number of for loops was increased and the average subsystem processing time measured. This data was collected using the data acquisition system and was analyzed using SIMplotter. This process was continued until 4 simulation frame overruns occurred. For the quad core system each core reached 140k loops (560k loops total) before 4 overruns occurred.

For this study a 2.4 GHz Intel based single core rtX was used as the system to be “enhanced” with additional processing power. An ADvantageDE project was built with the same C code model assembly to exercise the single core. The number of loops was increased until 4 simulation frame overruns occurred. The single core system number of loops was increased to 370k loops before 4 overruns occurred. In the best case the combined system has a potential of more than doubling the processing power.

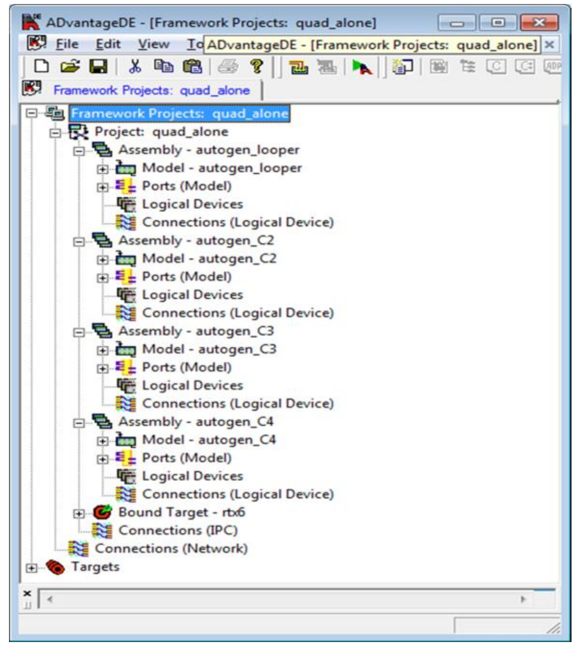

Figure 5. ADvantageDE project to test all 4 cores in quad core system.

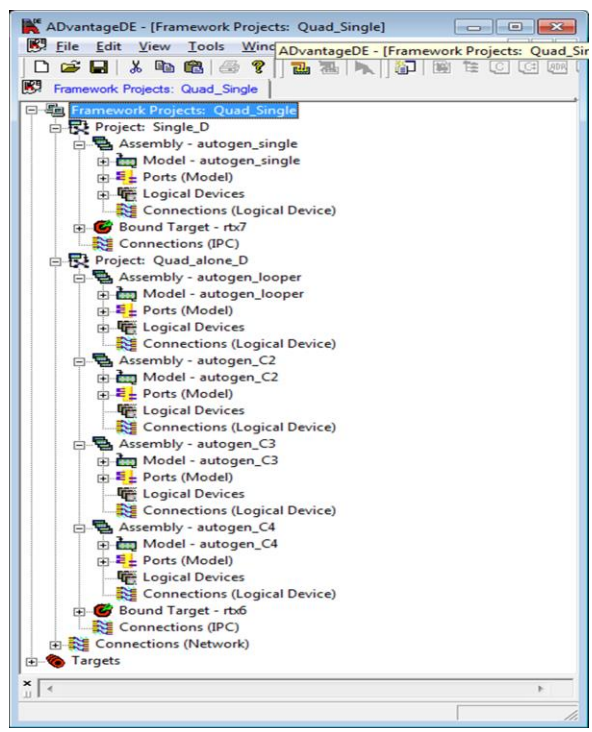

Figure 6. ADvantageDE Distributed project for combined system.

The next step of the study was to combine these two systems into a distributed simulation. Fig. 6 shows the ADvantageDE project. Fig. 7 shows the allocation of the model assemblies on the two systems, the single core and the quad core. The same C code assemblies were used to quantify processing power in the distributed system. The loop counts were increased until 4 or more simulation frame overruns occurred. In the distributed simulation on the single core system the loop count was reduced to 340k loops. In the quad core system the primary core was dedicated to communicating with the single core system, so the loop count on that core was set to 0. The remaining core loop counts were increased to 100k before overruns exceeded 4. In the distributed simulation the total loop count of the system was 640k, an increase of 270k over the original single core system, an increase of 73%.

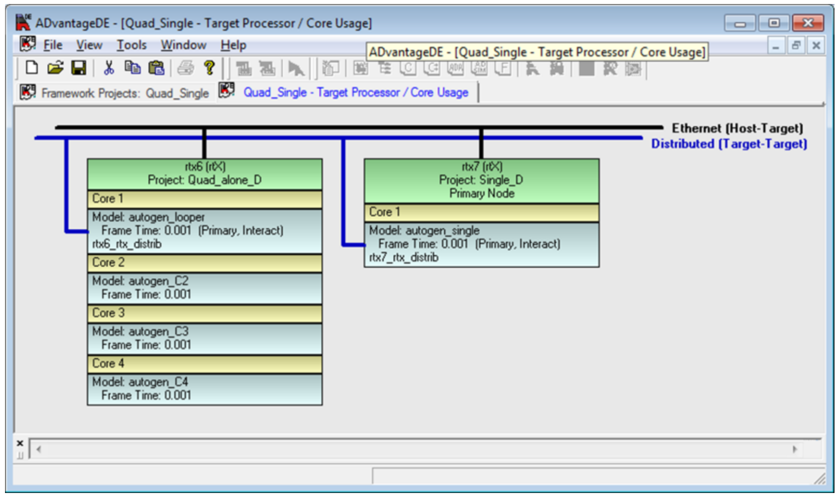

Figure 7. Target processor and core usage for single core and quad core systems.

VI. Modeling and Process Changes

In the ADvantage Framework projects can contain multiple assemblies. Assemblies are model elements that are connected together using IPC or network IPC. Using IPC introduces a single cycle delay in the data transferred from one assembly to another. In a single core rtX, the model can consist of a single assembly minimizing this delay. In a distributed simulation model assemblies are used to assign different portions of the model to different cores and projects are used to assign model assemblies to different nodes. Fig. 7 shows a distributed simulation with one model assembly assigned to the single core of rtx7 and 4 more model assemblies assigned to the 4 cores of rtx6. In this distributed model some latency will be introduced over a single assembly on a single core. Part of the design would be to determine whether this delay would be significant or not.

Adding another node with multiple cores to a simulation system changes the design process. First, the system being modeled has to be broken into model assemblies and the assemblies grouped into projects. In many cases the system itself will suggest logical decoupling of the model into assemblies. The assemblies and projects then have to be assigned to nodes and cores. For this task the statistics view in ADvantageVI provides timing data to aid in assigning assemblies and projects to the available resources.

Once the distributed system has been configured the process of running the simulation is similar to that of running a single node simulation. This makes the transition for the end user almost seamless.

VII. Conclusion

In this study a COTS computer was acquired, programmed with a commercial operating system, run-time services, and configured to be used in a simulation system based on ADI's ADvantage Framework. Testing showed in the best case the processing power of a single core system could be increased by 150%. Testing the distributed simulation showed the actual increase to be closer to 73%. The distributed network and the host communication were done over the same Ethernet network in this simulation system. Separating these networks could show additional improvement. Changing the simulation network to use SCRAMNet for the distributed network would also show additional improvement.

Modeling changes would be in the area of breaking the model into assemblies to be assigned to the additional computation resources. The Statistics View in ADvantageVI simplifies this task with timing data to help make design decisions. Once the distributed simulation is configured, running the simulation is similar to running a single node simulation with minimal impact on the end user.

In conclusion, processing power of existing simulation systems can be increased with the addition of COTS computers and the use of distributed simulation. With today's variety of form factors for COTS computers, a COTS computer can be found to fit nearly any equipment rack, even those with little to no space.

References

[1]James, S., Savaglio, C., "The Expanded Reach of Simulation Based Aircraft System Verification and its Software Capability Requirements", WWW.ADI.COM, Applied Dynamics International, Ann Arbor, MI, (unpublished).