LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-236998

CORBA-Based Distributed Software Framework for the NIF Integrated Computer Control System

E. A. Stout, R. W. Carey, C. M. Estes, J. M. Fisher, L. J. Lagin, D. G. Mathisen, C. A. Reynolds, R. J. Sanchez

December 5, 2007

Sixth IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research
Inuyama, Japan
June 4, 2007 through June 8, 2007

Disclaimer

CORBA-Based Distributed Software Framework for the NIF Integrated Computer Control System

E.A. Stout*, R.W. Carey, C.M. Estes, J.M. Fisher, L.J. Lagin, D.G. Mathisen, C.A. Reynolds, R.J. Sanchez

Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, USA

Abstract

The National Ignition Facility (NIF), currently under construction at the Lawrence Livermore National Laboratory, is a stadium-sized facility containing a 192-beam, 1.8 Megajoule, 500-Terawatt, ultra-violet laser system together with a 10-meter diameter target chamber with room for nearly 100 experimental diagnostics. The NIF is operated by the Integrated Computer Control System (ICCS) which is a scalable, framework-based control system distributed over 800 computers throughout the NIF. The framework provides templates and services at multiple levels of abstraction for the construction of software applications that communicate via CORBA (Common Object Request Broker Architecture). Object-oriented software design patterns are implemented as templates and extended by application software. Developers extend the framework base classes to model the numerous physical control points and implement specializations of common application behaviors. An estimated 140 thousand software objects, each individually addressable through CORBA, will be active at full scale. Many of these objects have persistent configuration information stored in a database. The configuration data is used to initialize the objects at system start-up. Centralized server programs that implement events, alerts, reservations, data archival, name service, data access, and process management provide common system wide services. At the highest level, a model-driven, distributed shot automation system provides a flexible and scalable framework for automatic sequencing of work-flow for control and monitoring of NIF shots. The shot model, in conjunction with data defining the parameters and goals of an experiment, describes the steps to be performed by each subsystem in order to prepare for and fire a NIF shot. Status and usage of this distributed framework are described.

Keywords: National Ignition Facility, NIF, CORBA, Integrated Computer Control System, Eric Stout, Lawrence Livermore National Laboratory, Object-Oriented software

* Corresponding author. Tel: +1-925-423-8863;
Fax: +1-925-422-1930
E-mail address: **stout6@llnl.gov** (E. A. Stout)

1. Introduction

The ICCS is a distributed, layered, object-oriented control system that employs a framework of reusable software to build uniform programs to satisfy numerous functional requirements for NIF [1]. ICCS applications are developed using Ada95 or Java,

CORBA, and object oriented programming. Ada is used to implement most of the control system semantics. Java is used for the production of graphical user interfaces as well as some of the central services; some new controls applications are also being written in Java. CORBA provides location- and language-transparent distributed communication utilizing TCP/IP transport.

NIF is comprised of 192 laser beams, which are divided into 24 essentially identical 8-beam "bundles." The control system computers and processes are likewise largely partitioned by bundle, with the exception of central services and applications controlling front-end, target area, and industrial controls systems that are not replicated. This partitioning is a key element in assuring the scalability of the control system.

The top layer of the ICCS bundle-based architecture is the shot automation system. Shot Director software manages the progress of a NIF experiment through a sequence of states. In each state, Collaboration Supervisors, one for each bundle and one for shared, non-bundle resources, manage the workflow of a sequence of steps defined in a shot model. These steps, in turn, are distributed to subsystem shot supervisors for execution [2].

Operational experience and analysis indicates that ICCS will scale successfully to full NIF operations. A full-power six-bundle shot was performed in late 2006, and preparations are underway for the first low-energy twelve-bundle test.

2. Bundle-Based Architecture

The NIF physical organization lends itself to a distributed, component-based communication architecture. Control system processes and computers are organized by bundle to achieve better parallelism, performance, and reduce the impact of localized failures. This is referred to as "bundle based partitioning" and has no impact on framework or application layer software due to the location independent features of the CORBA based inter-process communication architecture. This ability to independently organize the hardware architecture without impact to the software is one of the significant benefits of CORBA and the design of the ICCS software framework.

ICCS employs a layered, client-server computer architecture to control the NIF. Each NIF bundle is supported by ~25 front-end processors (FEPs) connected to various laser diagnostic and control components. Additional FEPs control and monitor common components that are not bundle based, such as target area systems. Each bundle has a dedicated Unix server on which the supervisory and shot control applications for that bundle are run. Additional servers support central service applications and non-bundle based supervisors. The control room contains Windows consoles dedicated to functional subsystems; GUIs run on these consoles and connect to the supervisory and FEP layers as needed. At full scale, the ICCS will be distributed among approximately 800 computers running 1500 processes containing 140 thousand individually addressable CORBA objects.

Following strategies of object oriented software development, similar software components are defined as classes, and these classes are instantiated for each occurrence of a NIF component in each laser beam. Control components consist of various actuators, sensors, and instruments used to operate and diagnose each NIF laser beam and its interaction with a target. NIF physical control components, as well as the supervisory objects that aggregate the status and control of those physical components, are represented by named CORBA software objects in the ICCS. The framework and application software combined have resulted in a design consisting of approximately 340 CORBA interface classes. The ICCS framework allows client software to obtain a CORBA reference to an object from its name, and CORBA transparently connects that reference to the software component object located in the processor that is connected to the physical hardware. The client need not have explicit knowledge of the process or processor on which the object is running. This gives computer/network location transparency to the ICCS application software.

There are a number of challenges to making such a distributed system robust and resistant to failure. A high number of distributed interfaces and objects performing concurrent, interdependent activities leads to non-deterministic messaging behavior and the potential for race conditions, distributed deadlock, or lost connections as a result of restarting a process. ICCS employs a variety of mechanisms under the broad heading of "connection

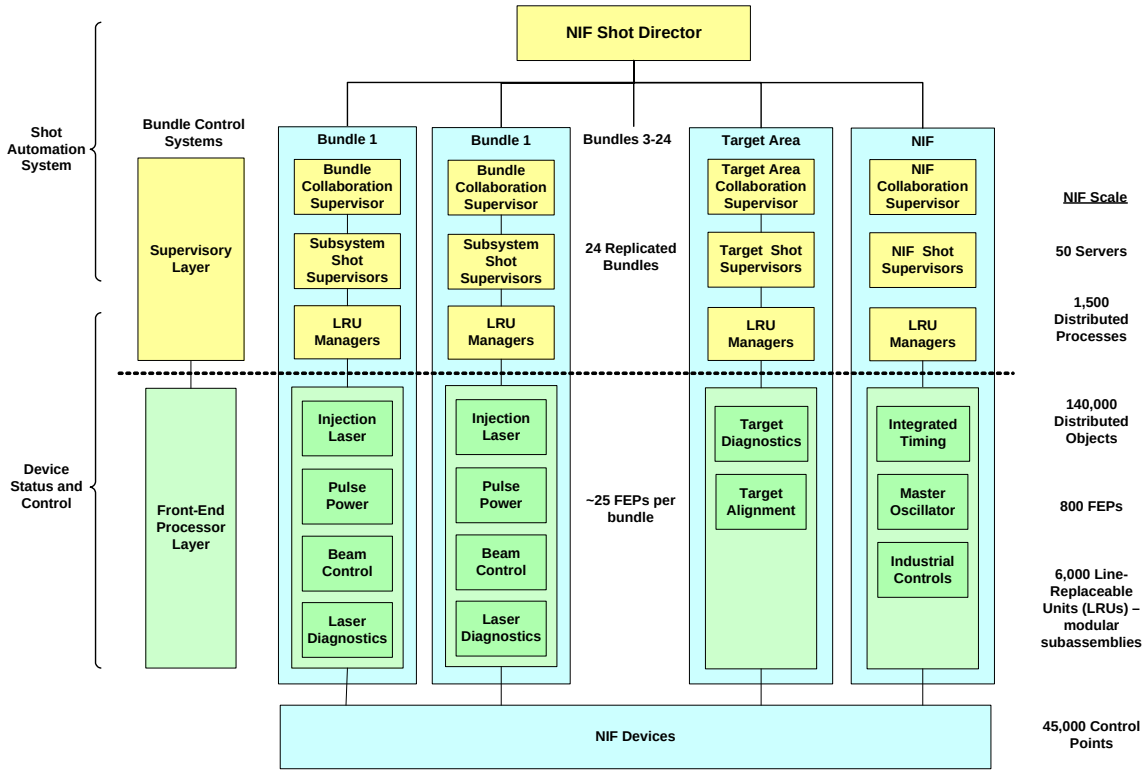

Figure 1 Bundle-based partitioning

management" to mitigate these failure modes, including interface decoupling, object reconnection, subscription management, process state heartbeats, status health heartbeats, and timed invocation. This framework approach has enabled successful detection, notification, and recovery from communication failures, providing common benefits to all ICCS application software.

3. Shot Automation System

Fundamentally, this large, distributed system exists for the purpose of enabling frequent, reliable NIF shots in support of experimental goals. The stated requirement for ICCS is to fire and diagnose each laser shot within 4-8 hours.

In a typical shot sequence, the laser is first setup to meet the experimental requirements. Automatic alignment software analyzes beam images and moves mirrors to align each beam accurately along the beam path and ultimately onto the target, and diagnostics are setup to obtain data from the shot. Then multiple low-energy "rod" shots, for which the power amplifiers are not fired, are taken, and settings adjusted based on the results, until the pulse shape and pre-amplified energy of every beam on the shot meet the experimental requirements. Finally, a high-energy "system" shot, in which the power conditioning system charges the main capacitor banks and fires the flashlamps that amplify the beams, is taken. Each shot is preceded by a 4-minute countdown during which an automated sequence of final actions and verifications is performed, and followed by data acquisition, analysis, and archival. Three layers of shot automation software orchestrate this intricate sequence: a Shot Director, Collaboration Supervisors, and Subsystem Shot Supervisors.

Upon selection of a predefined experiment by the Lead Operator, the Shot Director initiates calculations to determine operational settings and participation status for all of the NIF's components. It then manages a high-level state machine that shepherds the Collaboration Supervisors participating in the experiment together through each phase of the shot sequence. Once all Collaboration Supervisors have completed their activities for a state, the Shot Director moves them to the next state,

either automatically or at the request of the Lead Operator, depending on the state. Additionally, in the countdown and post-countdown phases, the Shot Director publishes clock ticks and manages any holds issued when problems occur. Two seconds before the shot (T-2), the Shot Director hands off control to the Integrated Timing System, which issues precise triggers to fire the laser so that all beams arrive at the target simultaneously.

There is one Collaboration Supervisor per bundle, plus two for non-bundle based systems. The Collaboration Supervisor is a workflow engine, coordinating all of the activities performed by the Subsystem Shot Supervisors in its area. When the Shot Director initiates a new shot state, the Collaboration Supervisor queries the database for the workflow model for that state. This model defines the high-level functions to be performed by each Subsystem Shot Supervisor, as well as their order and interdependencies. These high-level functions are referred to as "Macro Steps." The ordered list of Macro Steps for a single Subsystem Shot Supervisor is its "line of execution." Macro Steps may have as preconditions the completion of other Macro Steps in different lines, or even in different collaboration spaces. During countdown and post-countdown, a Macro Step may also have a clock tick as a precondition (e.g., do not execute until T-25, or 25 seconds before the shot). The Collaboration Supervisor manages all of its lines of execution in parallel, issuing each Macro Step when its predecessors have completed and all of its preconditions have been met. All of the lines of execution and their preconditions comprise a collaboration graph, which is presented on a GUI and updated with the status of each Macro Step so that the Lead Operator may observe the progress of the shot state. If a Macro Step fails, the Collaboration Supervisor allows the operator to select a reentry point at or prior to the point of failure, typically after the failure has been addressed. The Collaboration Supervisor then analyzes the interdependencies and "backs up" other lines of execution as necessary. Once all Macro Steps for the state are completed, the Collaboration Supervisor notifies the Shot Director that it has completed the state and is ready to proceed.

Roughly 10 Subsystem Shot Supervisors report to each Collaboration Supervisor. They are responsible for executing Macro Steps upon request. Each Macro Step is executed in four phases; the contents of each phase are defined in the database. First, a "Pre_Done Check" tests the exit conditions of the Macro Step to determine whether it needs to be performed at all. Second, a "Ready Check" verifies that the necessary equipment is in an appropriate state for the Macro Step to be performed. Third, the "Perform" phase executes the individual steps and sub-steps defined for the Macro Step. Finally, the "Final Done Check" verifies that the exit conditions of the Macro Step are met. All of these operations are described primarily in terms of setpoints. A setpoint is a name associated with a position or state of an ICCS component, which typically corresponds directly to the position or state of the physical device or collection of devices represented by that component. The three "Check" phases of a Macro Step typically consist of verifying that each of a set of components is either not participating in the shot, or is at the specified setpoint. The "Perform" phase commands components to their shot setpoints, and in some cases performs more complex behavior by executing a script.

The Shot Automation System provides a highly flexible framework for performing NIF shots. The specifics of a model or a Macro Step may be modified in the database without requiring modification of code. Different shot models may be defined to describe procedurally different kinds of shots. The initial experiment goals specify participating beams, energies, pulse shape, required diagnostics, etc. Derived goals establish settings required to accomplish the experiment goals. Based on these goals, database calculations determine the whether or not each component in the system is needed to participate in rod or system shots, as well as whether each Macro Step in the model is applicable (i.e., has any work to do) for the shot. A single shot model, combined with these participation calculations, may be used to perform a wide variety of experiments.

4. Scaling to 192 Beams

The ability of ICCS to scale successfully, to operate a fully-commissioned, 192 beam NIF is fundamentally important.

The bundle-based architecture is a key component for ensuring scalability: by deploying all of the controls for a single bundle on dedicated computers, and demonstrating that one bundle can be operated successfully, the ability to successfully operate the other 23 bundles by merely replicating the computers is effectively assured.

The remaining potential issue is the performance of the central services accessed by all bundles. In order to address this concern, an analysis of the central services' performance for a single bundle shot has been performed, and the results extrapolated to 24 bundles. Data on CPU utilization, memory utilization, and CORBA message traffic were included. This analysis indicates that ICCS will scale successfully to 192 beams.

In December 2006, a 960kJ 48-beam shot was fired, demonstrating the considerable capabilities of both NIF and ICCS on the largest scale to date. More recently, testing of simulated 96-beam shots has revealed the need to tune some CORBA configuration parameters to accommodate the increase in connections and concurrent operations. The first online 96-beam tests will take place later in 2007.

5. Summary

The ICCS provides a reusable, extensible, object-oriented framework for the development of controls applications. The framework and applications utilize CORBA for language- and location-transparent distributed communications. The framework provides a suite of connection management tools to address many of the complexities of a large distributed system. The bundle-based partitioning of ICCS applications mirrors the repeating bundle structure of the NIF to allow the control system to scale by replication.

The shot automation system leverages the ICCS framework and offers a highly flexible, data-driven mechanism for performing NIF shots that achieve experimental goals and collecting, analyzing, and archiving the results. The Shot Director manages the shot sequence through a series of high-level states; Collaboration Supervisors manage workflow as described by a shot model, read from the database, for each state; and Subsystem Shot Supervisors execute the Macro Steps specified in the model to setup the laser for the shot.

Operational experience performing multi-bundle shots, up to 48 beams, along with an analysis of the performance of the framework's central services extrapolated to 192 beams, provide confidence that ICCS will scale successfully to operate the full 192-beam NIF.

References

[1] R.W. Carey, et al, "Status of the Use of Large-Scale CORBA-Distributed Software Framework for NIF Controls," ICALEPCS 2005, Geneva, Switzerland, October 2005.

[2] L. Lagin, et al, "Shot Automation for the National Ignition Facility," 2005, Geneva, Switzerland, October 2005.

* This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.