NOVA: Networked Object-based Environment for Analysis

P.Nevski, A. Vaniachine, T. Wenaus[1]

[1] Brookhaven National Laboratory, Upton, NY, USA

Abstract

NOVA (Networked Object-based enVironment for Analysis) is a project to develop distributed object oriented physics analysis components, easily adaptable to different experiments. A useful set of tools has been developed, tested and deployed in prototype and production settings. A job configuration manager uses a framework's scripting interface to provide web-based editing, submission and cataloguing of analysis jobs, both user-level and experiment-wide, centrally managed in a database. A client/server system distributed over compute nodes provides job submission and monitoring across facilities, which may span several sites. A file catalog records production relationship of data files generated by an experiment. A CORBA interface to the file catalog for the Grand Challenge services has been implemented. NOVA provides database tools for geometry and parameter object storage. A NOVA web-based browser navigates a relational database storing hierarchically structured dataObjects. Clients may access database information from the code or through a CORBA-specified interface. NOVA components have been tested and deployed in the STAR (RHIC) and ATLAS (LHC) environments.

keywords Distributed computing, CORBA, database, MySQL, STAR, ATLAS

1. Introduction

The size and geographical distribution of current and next generation HENP experiments and their computing infrastructure makes support for distributed computing capability within the analysis software essential. At the same time, the rapid growth in networks capacities and technologies now underway makes a powerful distributed computing capability both possible and practical to a degree not seen in previous generations of HENP experiments. The computing infrastructure, which was centralized in the past, is now undergoing the trend to become more and more distributed[1]. The NOVA project[2] is building a toolset for a distributed computing in the Object Oriented analysis environments, adopted by forthcoming experiments.

Typically, an analysis framework steers a data analysis process from algorithm development through data selection, retrieval and filtering based on physics features, to the final result analysis, often in both interactive and batch environments. Many experiments have already established or are developing object-oriented frameworks. The NOVA project does not reinvent or evolve existing analysis frameworks, but rather provides new capabilities in these areas via modular components with application-neutral interfaces. They can be used in isolation to extend the functionality of the existing analysis systems. NOVA components will enable a thin local client to use an experiment analysis framework, providing transparent access to data and computing resources at a remote center.

While an early integration and prototyping of components is done in existing experiments, NOVA design and implementation is generic and applicable to the OO analysis environment of

---

[1] Amazingly, for experiments the trend is reverse — they are now became centralized, while in previous generations experiments were more distributed.

[2] `http://www.usatlas.bnl.gov/computing/nova`

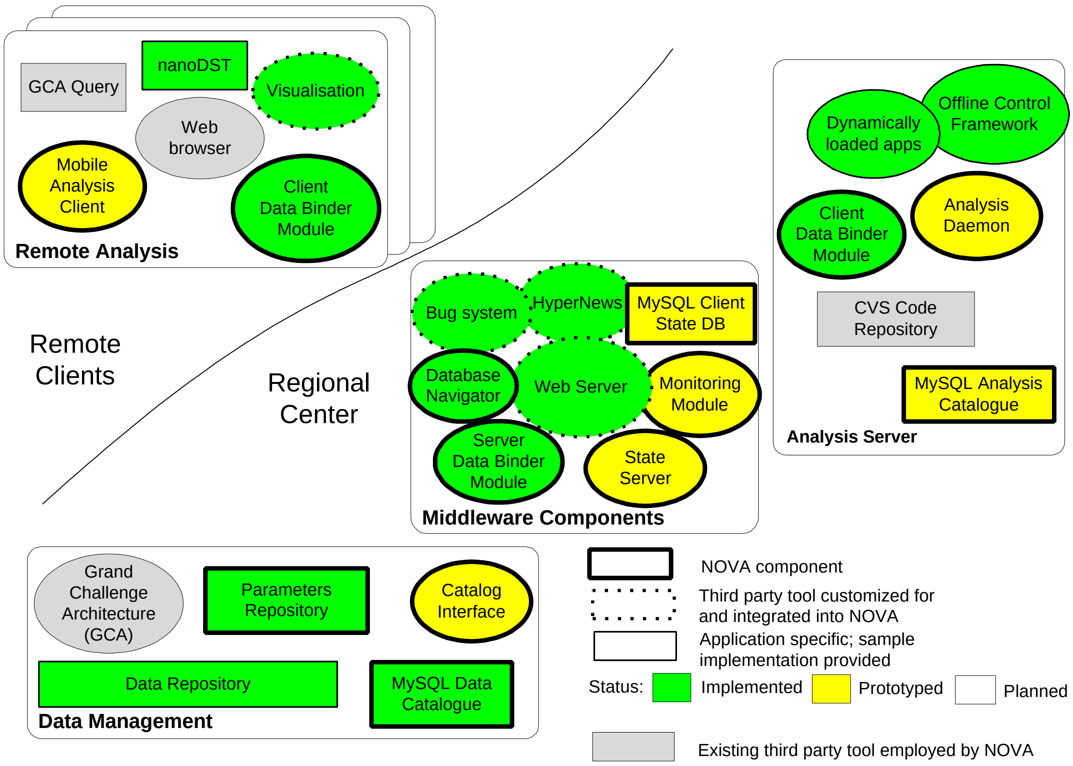

Figure 1. NOVA Architecture.

any experiment. We focus principally on supporting C++ based analysis, since C++ is the software language for the RHIC and LHC and other large experiments.

NOVA is being developed using an iterative process driven by user participation and closely coupled to prototyping in real experiments. Because NOVA developers are STAR and ATLAS collaborators these experiments have served as convenient test beds. During the first year of the project, all key components of the architecture have been prototyped or implemented and several have been deployed in production. We plan in the year 2000 to complete the implementation guided by application and prototyping in STAR and ATLAS, and establish NOVA as an experiment-independent toolset for distributed analysis.

2. Architecture and technologies

The NOVA architecture (Figure 1) consists of independent interoperable components designed for flexibility and ease of reuse. These components are distributed among the four domains:

- **Data management domain** – data and file catalog databases and associated interfaces. Central data and file catalogues with associated tools for controlling and monitoring data locality, data browsing, and evolving data models (schema evolution) for analysis.
- **Analysis server domain** – centralized analysis production and software management. Dynamic customization of a stable core analysis framework through shared libraries, with management of 'software signatures' for assured reproducibility of analysis environment.

- **Mobile analysis domain** – analysis client and associated tools for distributed analysis. A mobile analysis client served by a central analysis server and associated monitoring and control tools for physics analysis in a distributed, heterogeneous computing environment.
- **Web middleware domain** – communication and data exchange infrastructure. Integration of software distribution, management and version coordination tools with a mobile analysis client, problem reporting system, code navigation system, and discussion system to serve a widely distributed community.

Existing experience and the evolutionary path of HENP computing have guided the requirements to be met by tools and technologies employed within NOVA components. They should be free or nearly so, should be widely used standards with good support and should be known within the HENP community. Following these requirements we have adopted a set of tools (among several other open software tools) for application in the development of NOVA:

- **MySQL** – open Software relational database. Data catalogue, event store navigation, mobile analysis client state persistency, software signature management.
- **Apache** – open Software web server. Distributed client/server communication between mobile analysis client and analysis server. Web-based control and monitoring.
- **Perl** – Unix scripting and web development tool.
- **CORBA** – distributed inter-process communication.
- **ROOT** – object oriented toolkit for HENP analysis. Baseline implementation layer for analysis server framework, client analysis tools and data storage (optional, experiment-dependent component). Not a required NOVA component.

3. Implementation details

In this section we describe some NOVA solutions in the data management domain for problems typical in HENP computing, namely, how to support evolution of user code and how to support data workflow in an experiment, i.e. evolving collection of data files with their relationships.

3.1.1. Schema evolution for parameter database

In a research environment, users frequently come up with new ideas, not foreseen at the beginning. A typical outcome would be that a user modifies the structure of an object in his application, and the application saves a new object in the persistent store (database). Remote applications, unaware of the new functionality, may request the object in the old format. To simplify the schema evolution, NOVA provides forward compatibility to applications by implementing a dynamic object request broker scheme. In this approach the application complements a user request (name, time, selectors...) for dataObject, with a dictionary describing the dataObject, as expected by user code. In return, the database server supplies an available dataObject together with its own dictionary. A NOVA component – the Object Request Broker module – converts the database dataObject according to the application dictionary into a user expected type, when needed.

To achieve independence on a specific database choice, the NOVA access architecture is multi-layered. The inner "physical" layer is implemented by a MySQL server, while the "logical" database layer is supported by NOVA (Figure 2). It provides a capability of storing simple objects in a relational database. The upper layer - object request broker - is responsible for the database view to applications in terms of simple objects – structures. In this approach a database and analysis applications are separated via a robust interface with built-in type checking. The database access is independent of the application code version – user can read new dataObjects with an old code and take appropriate actions. The same database information is provided to all applications (e.g., C++ reconstruction code, GEANT-based simulation code etc) and is available for navigation or queries via a web browser.

3.1.2. Cataloguing data workflow

Another problem often appearing in big experiments is a continuous evolution and migration of raw and in particular reconstructed data. The problem is complicated by increasing data volume, as well as by a complex relationship between different data collections and conditions they have been produced within. For this task NOVA streamlined the data model for a file catalog. To simplify a user query for a particular file, status of all kinds of files is recorded in a single database table – the *fileCatalog*. It keeps track of files and their relations to production conditions. To distinguish between different kinds of files, NOVA introduces a concept of the file "producer". In a typical experiment principal file "producers" may be: *daq* – data acquisition run, *sim* – simulation, and *job* – event reconstruction (production). A *user* producer, referring to a hand-made input with simulation, acquisition or reconstruction parameters, may further supplement this set. Since files themselves are stored in a central repository (e.g. HPSS), the *fileCatalog* table also keeps an information regarding the primary file location. A complete list of other instances (copies) of the same files is maintained in a separate *fileCopies* table. To interface a third party software (i.e the Grand Challenge services [1]), all this information can be routed to the client through a CORBA server.

To store information relevant to particular producers NOVA uses distinct database tables, e.g., *daqInfo*, *simInfo* and *jobInfo*. Every production cycle performed by a producer is represented by a single entry in the corresponding table. Since each producer may create several output files, all their records in the database refer to the same entry in the producer table. In this way, the *fileCatalog* table is used to resolve the one-to-many relationship. Since the production job can, in principle, have multiple files on input, a table *jobInputFiles* is used to resolve the many-to-many relationship. Typically, there are many production jobs that correspond to the same software release and the same set of input parameters. NOVA has one record of these options in a separate table, with every job record referring to its particular production options.

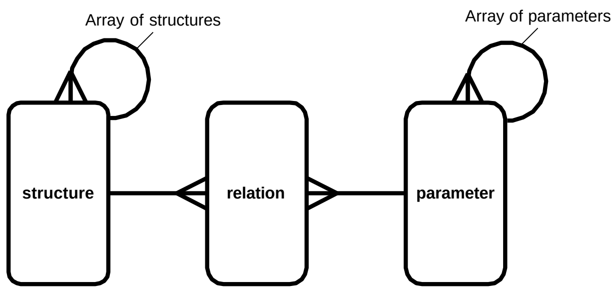

Figure 2. Data model for storing versioned objects in relational database.

In that way NOVA components enable different production jobs to share the common cataloguing and querying tool. For example, in the STAR implementation, a perl script creates a production job. The same script records in the database the job relationship to input files and output files history. The size of output files is initially entered as zeroes. Another perl script, which monitors the production job status, updates records in the database, e.g. inserts the actual size of the output files.

4. Summary

In its first year the project has succeeded in

- Implementing distributed computing and analysis tools in the areas of configuration and parameter databases, file catalogs and event meta-data databases, associated web-based browsers and management tools, and communication and distributed development tools.
- Implementing mobile analysis clients and a distributed analysis server system for job editing and submission, based on a centrally managed database and web-based clients.
- Implementing a machine, user, job and data monitoring system that provides the user a unified view of distributed analysis resources possibly spread across several sites.
- Deploying these tools in production or prototype form within STAR, and deploying parameter database tools in prototype form in ATLAS.

In the second year of the project we plan to

- extend the deployment and testing of NOVA components in STAR and ATLAS including scalability testing with management of multi-terabyte data sets and large scale analysis,
- refine component functionality and implementations based on experience from deployment and testing,
- complete the application-neutral interfaces of NOVA components to establish NOVA as an experiment-independent tool set for distributed analysis.

References

[1] Doug Olson et“New Capabilities in the HENP Grand Challenge Storage Access System and its Application at RHIC”, CHEP’00, Padova, (these Proceedings).