BdbServer++ A user-driven data location and retrieval tool

Alasdair Earl
University of Edinburgh
aearl@ed.ac.uk

Dominique Boutigny
IN2P3
boutigny@in2p3.fr

Adil Hasan
Stanford Linear Accelerator Center
hasan@slac.stanford.edu

on behalf of the BaBar Computing Group

Abstract

The adoption of Grid technology has the potential to greatly aid the BaBar experiment. BdbServer was originally designed to extract copies of data from the Objectivity/DB database at SLAC and IN2P3. With data now stored in multiple locations in a variety of data formats, we are enhancing this tool. This will enable users to extract selected deep copies of event collections and ship them to the requested site using the facilities offered by the existing Grid infrastructure.

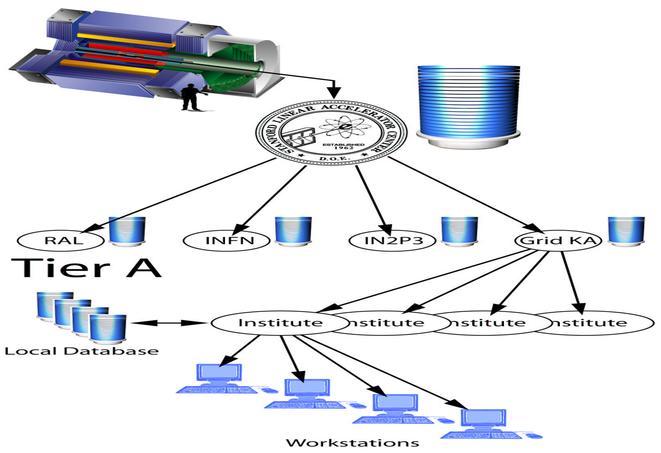
By building on the work done by various groups in BaBar, and the European DataGrid, we have successfully expanded the capabilities of the BdbServer software. This should provide a framework for future work in data distribution.

Introduction

There are two main data distribution issues facing BaBar today, bulk data distribution between Tier A sites and distribution of a selected set of data between Tier A sites and individual users. [1] looks at bulk data transfer between Tier A sites using the Storage Resource Broker. This poster discusses the issues facing developers in attempting to provide tools for users to locate, extract and retrieve the data they require.

The BdbServer++ project developed from the original work done by Dominique Boutigny and the increasingly distributed nature of the BaBar Computing Model. The original BdbServer has been used for several years to extract collections from the Objectivity/DB Object Database at SLAC and store them in temporary disk space to then be used for analysis.

The aim of BdbServer++ is to provide a Grid interface to the existing software and extending the functionality to allow users to make a deep copy of a pointer collection. We show how we have made use of current Grid middleware and how we plan to expand on this. We hope this will help other application developers in analysing some of the issues created by the Grid.



Background

The original BdbServer used email for its communication between users and SLAC. Users would sent an email to a central machine where their request was processed and an email was sent on completion. This was limited to a single site as security was minimal. However it was acceptable while all the necessary data and processing power was in a single location.

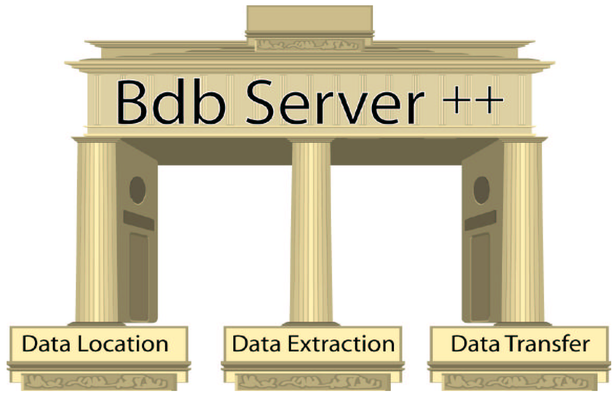
As Tier A sites developed in Europe, they started to specialise in specific tasks including storing certain data formats (Objectivity / KANGA). This has the obvious problem of either making users aware of what each site stores or creating a method of locating data for them. This is what we see BdbServer++ addressing in conjunction with other replica location services, replica catalogs and metadata systems.

Overview

BdbServer++ has three main tranches - data location, extraction and transfer. Data transfer is done using GridFTP, although any mechanism which supports 3rd party data transfer could be used. Data extraction, using the BdbCopyJob application, is available at all Tier A sites to copy collections from the main database. These two aspects are well know and supported.

Data location is a more interesting aspect of this project. Data may be in one or many locations, each having different levels of access reliability and restrictions. We currently only use data at Tier A sites but, as institutes increase the resources available and we improve our methods of ensuring data cannot be accidentally destroyed or corrupted, this may change.

Each site will hold a metadata catalog containing a list of collections at that site. Interfaces to this catalog will allow a user or application to obtain a list of collections available at their own and other sites by remotely querying the catalog. At a later point, it is hoped, a Resource Broker along with a replica location service could be used to locate and dispatch the deep copy jobs to the appropriate site. This will allow an instance of BdbServer++ to exist at each Tier A site and possible at Institutes as well. As these instances do not need to communicate with each other, we can provide users with an interface which can address multiple sites - leading to improved uptime and performance.



BBGUtils

While the main BaBar computing centres have adopted Grid middleware with gusto, the majority of Tier C (University Group Level) sites have yet to do so. Feedback from our intended user community demonstrated a pre-requisite requirement for a client package. This would provide an installer for the generic Grid components and the BaBar specific configuration necessary to submit requests and retrieve data. As this was a necessary precursor to the BdbServer++ project and has obvious advantages for other BaBar projects, we developed the BBGUtils package.

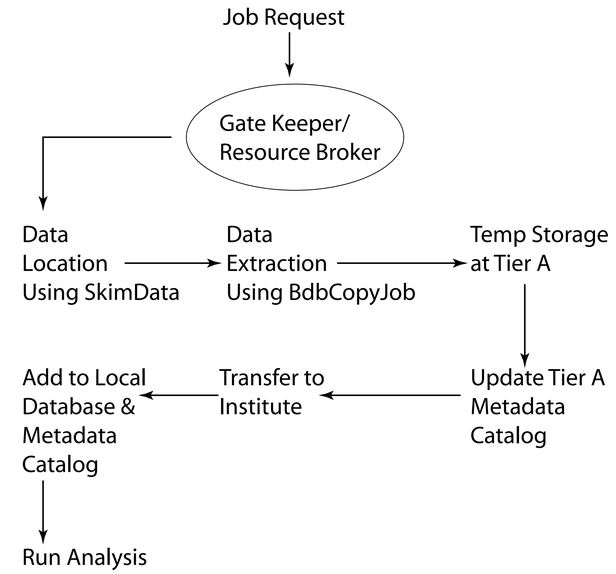
The ability to deal with multiple CAs had to be addressed, as did the ability to extend the package to include components such as the SDSC Storage Resource Broker Client and European DataGrid components. It was concluded that the BBGUtils installer should be aimed at people with limited computer knowledge - i.e. non-system administrators, and should avoid the necessity of installing components in system directories or under the root account.

BBGUtils was originally developed as a Bash Script and released in Autumn 2002. This had problems with non-Linux environments and was rewritten in Perl in the Spring of 2003. This new version provides the user with a variety of options for installing the required software. It is based on Globus Client 2.2.2 and is available from http://www.slac.stanford.edu/ ~aearl/BBGUtils.

Conclusions

BdbServer++ has highlighted some of the areas of opportunity for the current middleware. This has been fed back into the development process for future releases. The issue of data location is however, one where ideal solutions remain a distant prospect. Work on improving the registration of data throughout its lifetime will help and steps have been taken in this area. Further research into replica location services will allow us to decide on the best course of action.

This project has highlighted a need in the HEP community for tools enabling individuals to conduct curiosity driven research. By lowering the time and complexity of locating and accessing data we hope to give Physicists more time for analysis and research.



Future Work

BdbServer++ will continue to build on the work of BaBarGrid and the European DataGrid. We intend to evolve our software to take advantage of the functionality of resource brokers and replica location services as they become available and mature.

We hope that the work currently being done will aid the user community as well as the research groups and that this will prove to be a basis for future work in API and application development.

References

1. Distributing BaBar Data using the Storage Resource Broker, CHEP03
2. BdbServer - http://www.slac.stanford.edu/BFROOT/www/Computing/Offline/DataDist/BdbServer.html
3. BBGUtils - http://www.slac.stanford.edu/ ~aearl/BBGUtils/index.html
4. European DataGrid - http://www.eu-datagrid.org
5. BaBarGrid - http://www.slac.stanford.edu/BFROOT/www/Computing/Offline/BaBarGrid/

Acknowledgements

This work was funded in part by the Particle Physics and Astronomy Research Council under studentship PPA/S/E/2001/03338, the GridPP Travel Fund (www.gridpp.ac.uk), the Particle Physics DataGrid (www.ppdg.net), U.S. Department of Energy (www.energy.gov) and LAPP - Annecy CNRS/IN2P3 (www.in2p3.fr).