A New EPICS Archiver

N. Malitsky, D. Dohan

BNL, Upton, NY 11973, USA

Abstract

This report presents a large-scale high-performance distributed data storage system for acquiring and processing time series data in modern accelerator facilities. Derived from the original EPICS Channel Archiver, this version consistently extends it through the integration of deliberately selected technologies, such as the HDF5 file format and the RDB-based representation of the DDS X-Types specification. The changes allow scaling the performance of the new version towards the data rates of 500 K scalar samples per second and to provide a common platform for managing both the EPICS 3 records and EPICS 4 composite data types.

Rationale

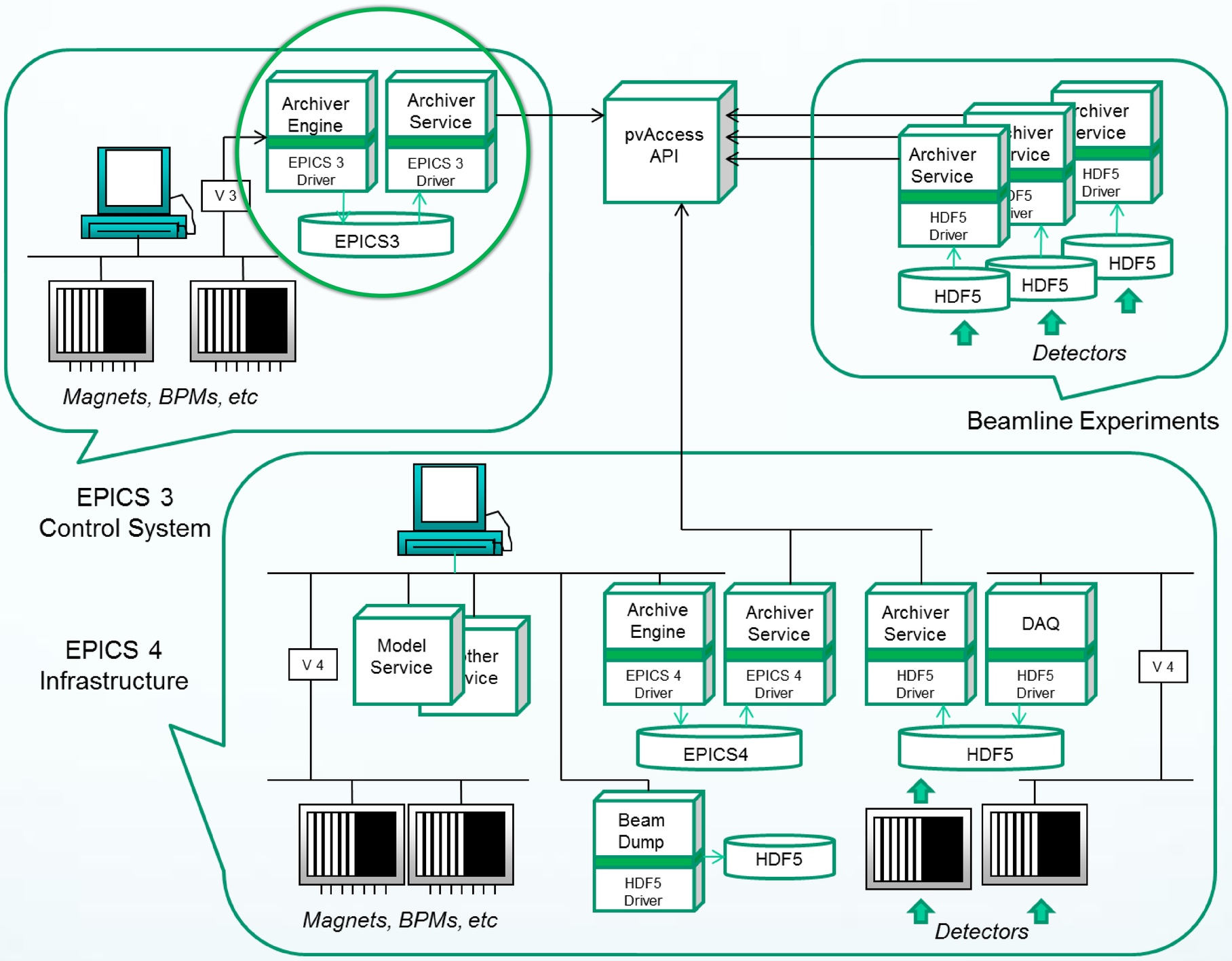
The scale, data rate, and complexity of new light source facilities introduce new challenges and demands for new approaches. Particularly, the BNL National Synchrotron Light Source II (NSLS II) shifts the frontiers of control systems towards millions of control process variables and streaming rates of up to one million events per second. Similar requirements are introduced by other accelerator projects. Furthermore, recent progress in the development of the EPICS v4 middleware triggers another request for supporting user-defined composite data types supported by the middle layer services. This change creates a natural path towards the consolidation of the control and experimental data management systems.

To address the data challenges of modern light source facilities and the new EPICS v4 infrastructure, we proposed an integrated approach derived from the original EPICS Channel Archiver architecture [1].

Integrated Approach

The development and application of the large-scale analytics-oriented data management systems is an emerging topic in academia and industry. Triggered by Google's web technologies, this domain represents an active factory for new products. Most of them are designed after Google's I/O stack: Google File System, Bigtable distributed storage system, and MapReduce processing framework.

Despite success in numerous projects, the web-oriented environment, however, cannot be directly applied to scientific applications. Analyses of different approaches confirmed the famous statement that “One size does not fit all” and lead us to a composite solution based on the integration of several open-source technologies. Figure 1 outlines the proposed data management system addressing the different versions of the EPICS control infrastructures end experimental facilities. In general, it is developed after the original EPICS Channel Archiver architecture with the integration of the HDF5-based backend and EPICS v4 distributed data services.



Data Store

The data store of the EPICS Channel Archiver is organized as a repository of data files maintaining time series of EPICS DBR data types. I/O access is asymmetric. The writing phase deals with the data sets of multiple channels accumulated during a short period of time. On the other hand, the data processing unit usually works with an extended history of a few channels. As a result, the original format of the data file was proprietary and highly optimized to address both requirements. The EPICS Channel Archiver format successfully employed with many EPICS-based control projects. However, it created significant constraints for adapting new user-defined data types introduced by the EPICS v4 applications and it was decided to evaluate a more general solution, employing the latest version of the Hierarchical Data Format (HDF5).

These studies clearly identified HDF5 as a winner among other alternative variants, including the file format of the SciDB database. The same task and associated studies, however, revealed a serious drawback of the present HDF5 specification, particularly, there was a lack of multi-file interfaces and services with one exception based on HDF5 proprietary external links. This problem has been naturally transformed into an advantage by integrating a repository of the HDF5 files with the original Channel Archiver indexing mechanism. As a result, this hybrid approach resolved the backward compatibility issues and provided a common platform for simultaneous processing of existing and new data files.

The extension of the HDF5 file format with the multi-file indexing service prompted consideration of a more general framework, data store middle layer, providing an efficient interface between the multi-file data repositories and processing engines. As noticed above, the time series of the EPICS v3 and v4 applications can be described using one HDF5-based data model: one-dimensional array of channel-specific structures. Following this approach, we complemented the original indexing service with a type service maintaining a catalog of the normative and user-defined data types.

The definition of a common type system is a difficult task involving a trade-off between the scope and complexity of numerous approaches. Recently, this issue has been addressed by the Extensible and Dynamic Topic Types (X-Types) specification developed in the context of the OMG Data Distribution Service (DDS). The specification provides the comprehensive type system model that overlaps the scopes of the EPICS v4 and HDF5 data types. As a part of the Archiver project, the X-Types specification has been mapped into the relation database representation and used for the registration of the EPICS DBR types.

Data Access Service

The data repository represents the backend of the archiver system that can be accessed with the additional data service. In the original version, this service was based on the XML-RPC protocol using the HTTP transport mechanism for exchanging XML-based messages. The recent EPICS v4 version provides another solution based on the novel concept of PV Data, a generic self-described dynamic data container. The transition to the EPICS-based integrated three-tier infrastructure introduced a significant advantage leading to the reimplementation of the original service in the new framework.

The data service interface consists of four essential commands and associated messages: archiver info, archiver names, channels names, and channel values. The fourth message represents the most complex case relied on a special container, a heterogeneous array of parameterized elements. In the PVData framework, this approach was implemented with another generic container, a dynamic structure of the self-described members. Other messages were directly mapped into the PVData basic data types, arrays of the homogeneous structures. Such one-to-one relationship facilitated the implementation of the new data service. Moreover, the same approach was applied in a straightforward manner to the clients such as the new archiver plugin of the Control System Studio.