### Abstract

The EPICS Archiver Appliance was developed by a collaboration of SLAC, BNL, and MSU to allow for the archival of millions of PVs, mainly focusing on data retrieval performance. It offers the ability to cluster appliances and to scale by adding appliances to the cluster. Multiple stages and an inbuilt process to move data between stages facilitate the usage of faster storage and the ability to decimate data as it is moved. An HTML management interface and well-defined customization hooks allow facilities to tailor the product to suit their requirements. The system has been developed at SLAC and MSU for about 2 years and is heading towards a production deployment at BNL. At SLAC, the system has significantly reduced maintenance costs while enabling new functionality that was not possible before. This paper presents an overview of the system and shares some of our experience with deploying and managing it at our facilities.

### Data Retrieval

- **Plugins for CS-Studio and ArchiView**: new bundled HTML5 viewer.
- **Support for processing the data during retrieval using post processors**.
- **Mean**, **Median**, **Standard deviation**, and **Others**: Use these same operators to **Precompute as part of ETL** – speed up response.
- **Decimate as part of ETL**: reduce data as ages.

### Installation

- **System requirements** - Server class machine + Recent versions of:
  - Linux
  - Java
  - Tomcat
  - A browser
- **Other requirements**:
  - MySQL for config.
- **Installation using**:
  - Puppet modules – makes installation a breeze.
  - Some scripts
- **Quickstart**: Quick setup to evaluate

### Administration

When users add a PV to the archiver, we:
- **Measure event storage rate**
- **Get RTVP, NAME, ADEL, MDEL etc**
- **Call installation specific policy (Python script)**
- **Policy makes configuration decisions**
- **Use capacity planning to assign PV to an appliance in the cluster**
- **Start archiving**

### Archive PV workflow

- **Possible to script the entire monitoring and administration of a cluster of appliances using Python scripts**

### Clustering

- **Short term store**: The most recent 2-3 hours of data; typically a RAM disk.
- **Medium term store**: The most 2-3 days of data; 15k SAS drives.
- **Long term store**: The rest of the data.
- **At SLAC**, this is bulk storage (with tape backups) that we rent; this is a GPFS filer system located elsewhere and is mounted over NFS.
- **At MSU**, this is a NetApp alliance with 2.8 TB of storage.

### Appliance

- **Each appliance has multiple storage stages** and **multiple processes**.
- **Many storage configurations are possible**. This is a typical configuration:
  - **Short term store**: The most recent 2-3 hours of data; typically a RAM disk.
  - **Medium term store**: The most 2-3 days of data; 15k SAS drives.
  - **Long term store**: The rest of the data.
- **At SLAC**, this is bulk storage (with tape backups) that we rent; this is a GPFS filer system located elsewhere and is mounted over NFS.
- **At MSU**, this is a NetApp alliance with 2.8 TB of storage.

### Figure 1: Each appliance has multiple storage stages and multiple processes

- **Many storage configurations are possible**. This is a typical configuration:
  - **Short term store**: The most recent 2-3 hours of data; typically a RAM disk.
  - **Medium term store**: The most 2-3 days of data; 15k SAS drives.
  - **Long term store**: The rest of the data.
- **At SLAC**, this is bulk storage (with tape backups) that we rent; this is a GPFS filer system located elsewhere and is mounted over NFS.
- **At MSU**, this is a NetApp alliance with 2.8 TB of storage.

### Figure 2: An installation is a cluster of appliances. Scale by adding appliances.

To this strip, all appliances use the same LTS. However, you can also have each appliance use a separate LTS. The architecture is shared nothing.

### Figure 4: Use traffic management between clients and appliances.

### Channel Archiver Integration

- **Transparency proxy for the ChannelArchiver or XMLRPC server**.
- **No need to migrate data to new format**.
- **However**, MSU has developed utilities to do so if desired.
- **Import ChannelArchiver XML config files**.
- **Basic functional requirements**:
  - a2aproxy
  - a2aproc
  - Use as a switchable proxy between a ChannelArchiver and an EPICS Archiver Appliance.