Fusion Engineering and Design xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Conceptual design of new data integration and process system for KSTAR data scheduling

Taehyun Tak*, Jaesic Hong, Kaprai Park, Woongryul Lee, Taegu Lee, Hyunsun Han, Giil Kwon, Jinseop Park

National Fusion Research Institute (NFRI), Daejeon, Republic of Korea

ARTICLE INFO

Keywords:
KSTAR
Fusion
Dataflow
Automation
Many-task
Big Data

ABSTRACT

The KSTAR control and data acquisition systems mainly use data storage layer of MDSPlus for diagnostic data and channel archiver for EPICS-based control system data. In addition to these storage systems, KSTAR has various types of data such as user logs from Relational Database (RDB) and various types of logs from the control system. A large scientific machine like KSTAR is needed to implement various types of use cases for scheduling data and data analysis. The goal of a new data integration and process system is to design the KSTAR data scheduling on top of the Pulse Automation and Scheduling System (PASS) according to KSTAR events. The KSTAR Data Integration System (KDIS) is designed by using Big Data software infrastructures and frameworks. The KDIS handles events that are synchronized with the KSTAR EPICS events and other data sources such as the rest API and logs for integrating and processing data from different data sources and for visualizing data. In this paper, we explain the detailed design concept of KDIS and demonstrate a data scheduling use case with this system.

1. Introduction

Systemically, a large scientific machine such as KSTAR [1] is implemented with a complex system design including heterogeneous hardware and software platforms. In other words, integrating the control and data systems is one of our major missions for the KSTAR operation and experiment. In order to integrate the systems, KSTAR adopted EPICS [2] as its main integrated control framework and a channel archiver for its storage. For diagnostic systems, we developed a standard framework [3] for the sequential archiving operation of digitized data with MDSPlus [4]. In addition, various type of file formats, file systems, and relational database mechanisms were used for its own purpose according to each type of data. The generated data from control and diagnostic system could be stored, analyzed, or refined in the various use cases. Since the first operation of KSTAR in 2008, its control and data storage systems have been stabilized and matured to perform experiments for the KSTAR mission.

On the other hand, the amount and type of KSTAR data has been increasing every year as the system is upgraded. Therefore, the complexity of data and its use cases are also increasing. Finally, a need has arisen for a specific system that can efficiently develop complex data use cases such as large amounts of dataflow processing, various algorithm processing, and visualization.

As the first step of our challenge for the next generation of data processing environment in KSTAR, we considered a new data orchestration scheme and prototyped a new data integration and process system named the KSTAR Data Integration System (KDIS). This system supports a wide range of KSTAR pulse operation automation and experimental result process with offline data processing (for non-real-time or soft real-time works). Therefore, the focus is on building a new frameworks that uses Big Data processing methods rather than the legacy process such as single process application with storage. The KDIS is an operation-related data-cycle ecosystem synchronized with KSTAR events (data), with a distributed computing system including general-purpose Big Data open-source frameworks and various libraries. In this paper, we will describe the development and demo results of the KDIS.

2. KSTAR data integration system

The main purpose of the KDIS is to establish a data orchestration system with a dataflow that covers the storage, processing, and servicing of generated and processed data from KSTAR.

For various KSTAR data-intensive applications and data scheduling use cases, an appropriate architecture design is necessary and our top priority. The Lambda architecture design paradigm [5] covers computing arbitrary functions on arbitrary data use cases by dividing the

* Corresponding author.
E-mail address: thtak@nfri.re.kr (T. Tak).

https://doi.org/10.1016/j.fusengdes.2018.01.015
Received 22 June 2017; Received in revised form 22 December 2017; Accepted 3 January 2018

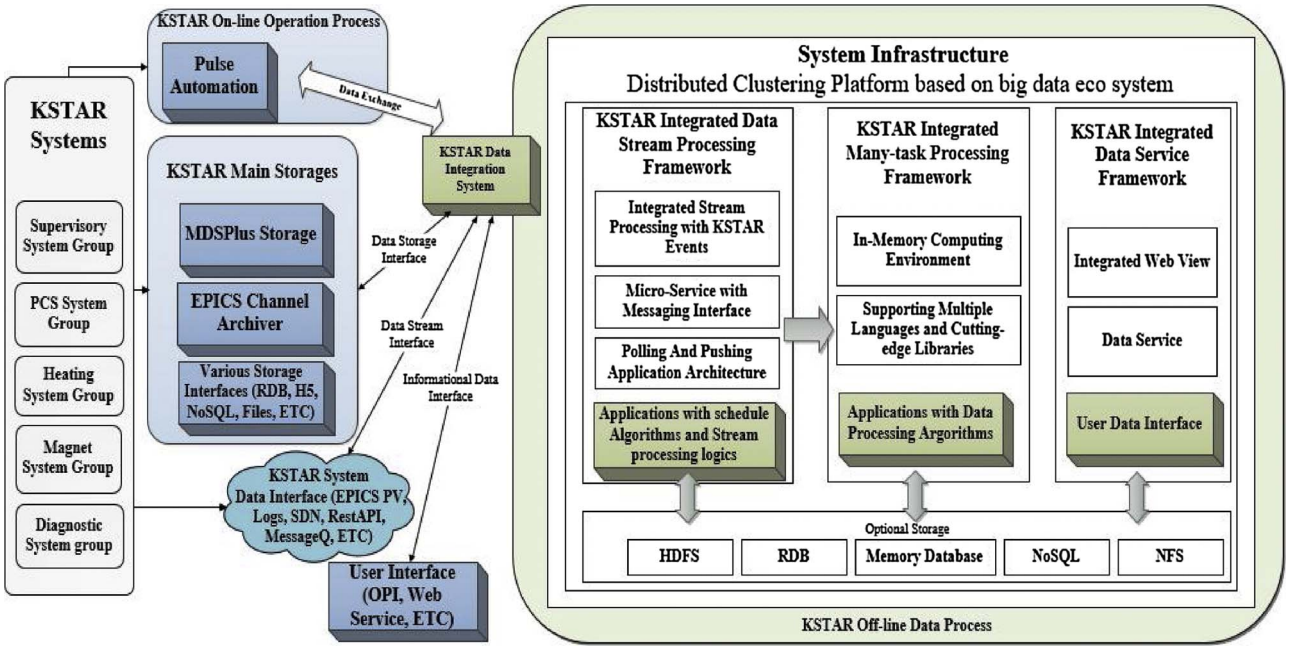

Fig. 1. The framework design of KDIS and its interconnection among KSTAR systems.

Table 1
The KDIS System Hardware Configuration – Clustered Servers (4EA).

Component	Specification
CPU	Intel (R) Xeon® CPU E5-2640 v4 2.4 GHz * 2
RAM	128 GB RAM
Storage	500 GB SSD * 2

Table 2
The KDIS System Software Configuration.

Component	Specification
OS	CentOS 7.2
Resource Scheduler	Hadoop Yarn
Stream Processing F/W	Spring Cloud Dataflow (SCDF)
Many-Task F/W	Apache Spark
Web Application Server	Apache Tomcat (Spring MVC + Hibernate)
Data Storages	HDFS, Geode, HBase, PostgreSQL
Library Environment	Many types of open-source or internally developed libraries are used by types of applications

data related mechanisms into three layers: the batch layer, the serving layer, and the speed layer. The KDIS is inspired by the Lambda architecture design pattern on the point of handling arbitrary data use cases dividing mechanisms into the layers, and redefines the role of each layer according to the KSTAR dataflow.

To simplify the functionalities, the KDIS is defined as three frameworks by architectural design: stream processing framework for applications implementing stream processing algorithms such as event processing and reactive programming, many-task processing framework for tasks implementing data analysis logics with batch data processing, and service framework supporting data view to users. The designed architecture and interconnection between KSTAR data and the KDIS is shown in Fig. 1. KSTAR data interfaces are connected to the KDIS via a network with various interface libraries such as EPICS client, MDSplus client, memory DB, RDB, message interface and so on. Implemented applications on the stream processing framework could communicate the KSTAR system such as pulse automation system, data storages, and stream by polling and pushing mechanism. On the conceptual system, we developed the task scheduler as one of main features for deployment of jobs. This scheduler interfaces with EPICS data stream for event processing according to scheduling algorithms. It launches logics on tasks in the many-task framework depending on the condition. Processed data will be provided via a data service framework with visualization method.

For implementing functionalities of framework architecture, the KDIS is configured with Big Data open-source solutions to each frameworks. Tables 1 and 2 lists the major hardware and software components of the KDIS. The KDIS manages resources of clustered systems using a Hadoop Yarn scheduler [6]. The scheduler manage system resource with all applications running on the KDIS system. Each open-source solution has convenient mechanisms and features such as managing and monitoring applications, supporting in-memory computing, supporting convenient development, and supporting interfaces with a well-structured architecture. The KDIS takes advantages from well-developed Big Data open-source solutions. Detailed introductions of frameworks are described through this chapter.

2.1. Stream processing framework

The stream processing framework is mainly composed of a structure to deal with both of polling and of pushing stream processing applications by publishing, subscribing, and storing data. The stream processing applications are developed under the open-source framework Spring Cloud dataflow [7], which is a toolkit for building data integration and real-time data processing pipelines. Applications are developed as micro-services and connected to each other with pipelines which framework support. The micro-service applications could be developed with the role of source, process, or sink. This framework allows applications to support a various data-processing use cases, from import and export to event streaming and predictive analytics. This open-source framework supports the KDIS runtime environment of stream applications by powerful usability such as supporting stream design flow diagram GUI, deployment on Hadoop, program code minimization by using abstracted network program APIs, and more.

On this conceptual system, launching tasks with a task-scheduling algorithm is mainly developed for batch atomization to analyze data by KSTAR sequence. This function is accomplished in connection with a KSTAR data source such as EPICS process variable (PV). Through this functionality of stream application, KSTAR data will be scheduled by