ADAPOS: AN ARCHITECTURE FOR PUBLISHING ALICE DCS CONDITIONS DATA

J. Lång[1,2],
A. Augustinus[1], P. M. Bond[1], P. Chochula[1], M. Lechman[1,3], A. Kurepin[1,4], O. Pinazza[1,5]
(1) European Organisation for Nuclear Research Physics Department (PH), (2) University of Helsinki, (3) Slovak Academy of Sciences (IP SAS) Institute of Physics,
(4) Russian Academy of Sciences (RAS/INR) Institute for Nuclear Research, (5) Istituto Nazionale di Fisica Nucleare (INFN-Bologna) Sezione di Bologna

Background

ALICE Data Point Service (ADAPOS) is a software architecture developed for the *Online-Offline* (O^2) project of the *Large Ion Collider Experiment* (ALICE) of the *Large Hadron Collider* (LHC) of the *European Organisation for Nuclear Research* (CERN). The O^2 project involves extensive upgrades to ALICE as the interaction rates will increase by a factor of 100, during RUN3 period of LHC (to commence in 2021). ADAPOS is part of the pipeline for transmitting conditions data from *ALICE Detector Control System* (DCS) to the O^2 infrastructure, where it will be used in the reconstruction of physics data from the experiment.

Software Technologies

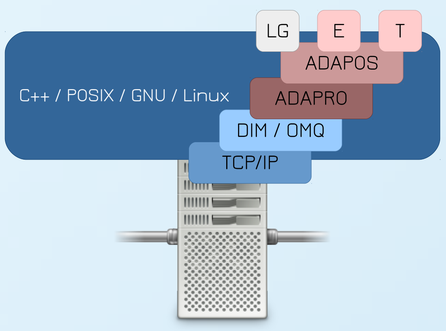

Figure 1: Stack of Technologies

ADAPOS applications are mainly targeted for the CERN CentOS (Linux) operating system. The programming language for the implementation was chosen to be C++14, due to the soft real-time requirements for the software. ADAPOS uses *Distributed Information Management* (DIM) protocol – also developed at CERN – and *0MQ* for networking. Other common functionality of ADAPOS applications was moved to the publicly available, open source (with CERN licence), *ALICE Data Point Processing Framework* (ADAPRO) during early stages of development.

Conditions Data

The basic datum of conditions is called a *data point*. A data point can contain a temperature or voltage reading, typically from a single device or channel. Data points are published as DIM services. ADAPOS converts data points into fixed-size binary records, known as *Data Point Composite Objects* (DPCOM). ADAPOS outputs DPCOMs using a 0MQ socket.

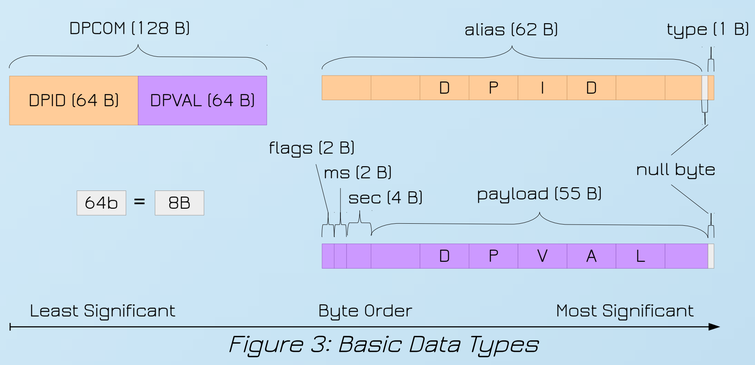

Figure 3: Basic Data Types

ADAPOS Applications

The two main ADAPOS applications are called (ADAPOS) *Engine* and *Terminal*. Engine runs on a server in DCS network, and obtains conditions data by subscribing to the DIM services. Engine sends DPCOMs to Terminal, using a 0MQ socket. Terminal runs on a *Front-Line Processor,* (FLP) facing both the DCS and *Data Aqcuisition* (DAQ) networks. It updates a segment of DPCOMs, called *Full Buffer Image* (FBI). An FBI represents a snapshot of the state of DCS. Terminal keeps sending FBIs to the DAQ readout application at regular intervals.

Figure 3: Basic Data Flow

ADAPOS Load Generator

(ADAPOS) *Load Generator* (LG) is an ADAPRO application for Linux, that publishes data points using DIM protocol, and updates them periodically with an adjustable delay between updates. It is used for simulating conditions data sources, during the development of ADAPOS software. The reliability and performance of ADAPOS was tested by running simulations with various number of load generators with different parameters. A load generator implemented in WinCC OA was also used during earlier stages of development.

Simulated Parameters

The main goal in the simulations was finding out the limits of ADAPOS performance and observing what happens when these limits are exceeded. The simulated parameters were the number of LGs, the number of data points published per LG, and the delay between data point updates generated in a single LG process. The impact of Linux scheduler and virtual memory locking was also tested using an application called *numactl*.

Simulation Results

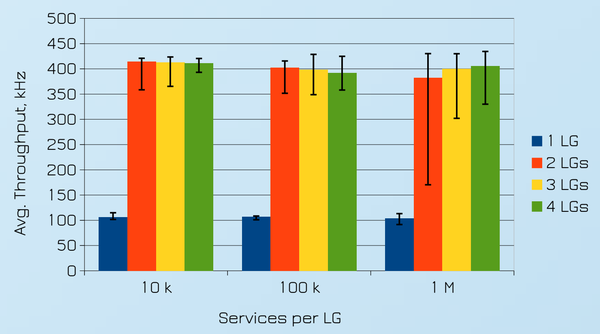

Figure 4: ADAPOS 1.0 Throughput Rate. The rate is about 25% lower for the latest ADAPOS version (4.1), because of increased software complexity.

When operating below the maximum capacity, the performance of ADAPOS was found out to scale roughly linearly. The number of data points per LG don't seem to have effect on performance, except for increasing the startup time of Engine proportionally. Also, the memory-footprints of ADAPOS applications don't increase over time, which implies the absence of serious memory leaks. *ADAPOS was found to operate with high level of reliability and stability* (apart from occasional problems caused by software bugs, that exist no longer), even during longer simulations that lasted for days on maximum throughput. *The simulations of ADAPOS applications also provide a case study for demonstrating the features of the ADAPRO software framework.*

Future Prospects

I*t was found that the Linux scheduler has a major impact on ADAPOS performance,* especially when it moves threads from one CPU socket to another, on a multi-socket system. Because ADAPOS is a input-output oriented system, performance improvements are hard to achive by compiler optimizations like loop vectorization. Instead, adjusting scheduling, paging and other such runtime system parameters, is likely to yield speed boosts more easily. Ideally, these adjustments could be implemented in ADAPRO framework itself, making it more attractive tool for general audience in real time parallel programming or physics control systems. Other possible future improvements include an unified asynchronous stream-like API to DIM and 0MQ access and a simpler thread model in ADAPRO, and a configuration database for ADAPOS that would replace configuration text files.

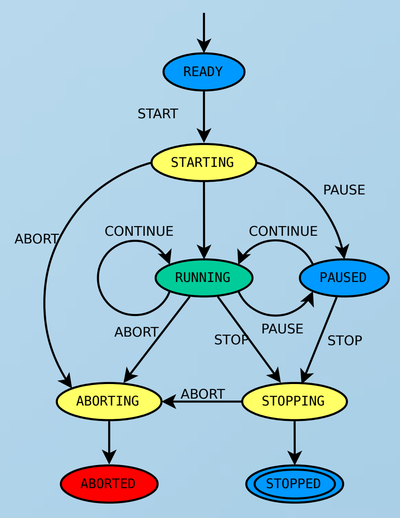

Figure 5: ADAPRO Thread Model

See Also

ADAPRO: https://gitlab.cern.ch/adapos/ADAPRO
DIM: https://dim.web.cern.ch/dim/
0MQ: http://zeromq.org/

Partners

ALICE
A JOURNEY OF DISCOVERY

UNIVERSITY OF HELSINKI

fuug