DABC as event builder at HADES experiment[*]

J. Adamczewski-Musch[1], *S. Linev*[1]
[1]GSI, Darmstadt, Germany

Introduction

The Data Acquisition Backbone Core (DABC) is a software framework for distributed data acquisition [1]. Due to its plug-in mechanisms it is extendable to various data formats and experiments. For several test set-ups of trb3 frontend readout, a DABC plug-in had already been developed that could receive and combine HADES *trbnet* data packets via UDP connections [2]. These developments have now been completed to cover the full functionality of the previous HADES event building software. This includes interfacing the EPICS based HADES control system, the ORACLE run statistic data base, and the RFIO tape storage of GSI. Initialization procedures of the DABC event builder processes have been fully integrated into the existing HADES DAQ configuration. Moreover, by means of DABC stream server socket the connection to HADES online quality monitoring analysis could be improved. In 2014 DABC was applied for data taking of several beam times at HADES experiment.

DABC integration to HADES

Figure 1 shows the integration of DABC into the HADES data acquisition system [3]. The digitized data from the front-end TRB boards is send via Gbit-Ethernet UDP to event builder processes that have been replaced by DABC. The shared memory interface to the existing EPICS control and run synchronisation system has been implemented fully compatible with the previous event building software *hadaq*, i.e. it is possible to operate the DAQ with “mixed” event builders (DABC and *hadaq*). To achieve such flexibility, the existing HADES DAQ set-up scripts have been extended in a way that each event builder process can be started with the same settings (receiving UDP ports, file output, etc.) either with DABC or *hadaq*. These parameters are passed from the startup script to the DABC process by shell environment variables, which are evaluated in the DABC xml configuration file generic for all event builders. On the other hand, DABC logging and debug output have been attached to the HADES logging facility based on Linux *syslogd*, so most important event builder messages are available here.

HADES data taking “run statistics” (start/stop time, number of events) is exported to ORACLE data base by means of intermediate text files written by the event builders. This data base interface has also been implemented in DABC. HADES event data is written in *hld* file format both to an array of local hard disks, and via Gbit Ethernet sockets to the RFIO tape library of GSI. The *hld* file output plug-in had previously been implemented to DABC [2] and has now been extended by the HADES requirements:

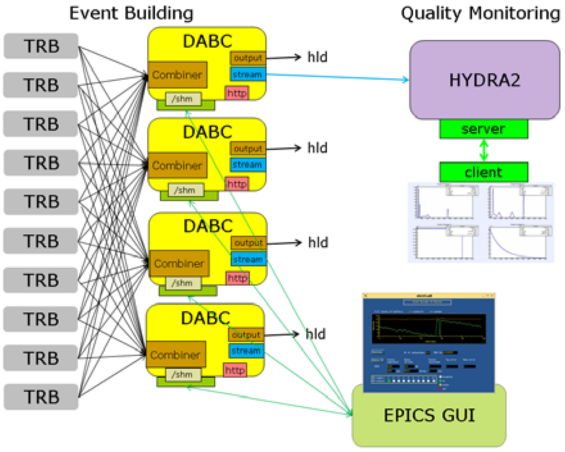

Figure 1: HADES data acquisition and quality monitoring with DABC event builders

- the file names must contain run id and event building node;
- files of all event builders must automatically close and re-open when the run id is changed, which is synchronized via EPICS;
- for load balancing of the local disk arrays, the destination partition for each new file is distributed to the event builders from a demon process via shared memory;
- the event builders may connect to the GSI tape storage by a dedicated RFIO interface library that has been embedded to DABC.

Improved online monitoring

The HADES Quality Analysis (QA) online monitoring needs a sample of event data to analyse and watch the detector states in a quasi live display (Fig.1). Formerly this was provided by temporary “small files” written by one of the 16 event builders and read back by the QA process via an NFS mounted disk. Since DABC can deliver event data to a client via an “MBS stream server” socket, this connection has now been implemented and used by the HYDRA2 QA process. So intermediate files and common disk access are not required anymore, and the *stream server* is available even if no event data is collected to disk. Additionally, several different analysis clients could connect to the DABC event builder and can receive the same event samples simultaneously. Moreover, stability and reconnect time of the QA process have been improved because such client/server socket connection lacks most problems of concurrent producer/consumer file I/O.

[*] PSP code: HADES 1.1.2.4

DOI:10.15120/GR-2015-1-MU-NQM-HADES-26

Beamtime experiences

In May 2014 a HADES proton beam time was performed to prepare and align for the subsequent pion beam times. Here for the first time one event builder process (EB5) ran with DABC, all others still with previous *hadaq* software and the old QA file export. In July 2014 the first HADES pion beam time block also applied one DABC event builder. However, here DABC was handling event builder node EB2, uniquely receiving all special calibration events and serving the QA process via *stream server* socket. Finally at the second "production" pion beam time in August/September 2014 [4], DABC was running all event builders. The performance limit without data loss of this system has been measured with a 25 kHz pulsed trigger at 29 frontend systems, feeding usually 3 active event builder processes only with full disk and tape file output. Here each event builder could handle data rates of 38Mbyte/s with a maximum CPU load of 78%. This test goes far beyond the typical observed pion beam event rates of < 2 kHz, so data taking with DABC worked well and was sufficient here.

From such beam time experience several improvements and bug fixes have been added, concerning e.g. EPICS data rate monitoring, debug output of missing data senders, no data buffer loss at long spill pauses, and run statistics export. Additionally, internal performance of DABC framework and plug-ins could be developed further in the scope of HADES application.

Conclusions and outlook

DABC has proved to be suitable for production event building and data taking of HADES sized experiments. Due to its flexibility it was possible to fully support all features of the former *hadaq* event builder software and control environment. Moreover, additional DABC features like event data *stream server*, or http control web server, have opened up further possibilities for online monitoring and control at HADES experiment.

References

[1] J. Adamczewski-Musch, H.G. Essel, N. Kurz, S. Linev, "Dataflow Engine in DAQ Backbone DABC" IEEE TNS Vol.57, No.2, April 2010, pp 614-617

[2] J. Adamczewski-Musch, S. Linev, E. Ovcharenko, C. Ugur, "HADES trbnet data formats for DABC and Go4", GSI Scientific Report 2012, GSI-SR2012-PHN-SIS18-ACC-41

[3] J. Michel et al., "The upgraded HADES trigger and data acquisition system", JINST 6 C12056, 2011

[4] HADES pion beam time 2014, this report

DOI:10.15120/GR-2015-1-MU-NQM-HADES-26