Controls

Session: 10 April 2023
Tutors: Alessandro Stecchi (Istituto Nazionale di Fisica Nucleare), Giampiero Di Pirro (Istituto Nazionale di Fisica Nucleare)

Food for thought

It is considered beneficial to provide the students of this course with a concise outline of what has been said and done during the laboratory session.
This outline contains food for thought that each student can use as they see fit: to better remember what was discussed in the introductory presentation, to dig into certain topics just for personal curiosity, or even to approach – in the future – an IT (Information Technology) topic with a different mindset.
In any case, this should be considered an informal document, and its reading or use is not required in any way (even though allowed) for the drafting of the activity report.

How the approach to computer systems development changes

In software development and hardware infrastructures design, we now have highly advanced technology compared to just a few years ago, and this radically changes both how applications are written and how computing infrastructures and platforms are designed.

Keywords: virtual machines, dockers, Kubernetes, orchestrators, high throughput networks and storage.

How the working method changes

Modern methodologies in the field of software development adopt innovative practices aimed at accelerating the release *values* to the customer. For example, the AGILE approach allows for working in successive iterations (sprints) with the release of new features at the end of each iteration. This method is mostly more effective compared to the traditional *waterfall* approach, where the product is released only at the end of a single development process consisting of a series of sequentially dependent actions.
All of this not only applies to software development but generally to the creation of anything: a system, an activity, etc.

Keywords: AGILE, Continuous Integration & Continuous Delivery, Project Tracking, DevOps, Quality, Documentation, Asset Management, OOD, OOP

Control systems specific characteristic

In this course, we focused on control systems based on control processes implemented in software and operated via computers. We also mentioned that, essentially, controlling a particle accelerator does not differ from controlling an industrial, medical, or other type of plants. On the other hand, we have also seen that there are different requirements for these different contexts that require optimization of the design aimed at making it suitable for operating the specific context.

Keywords: prognostic, AD (Anomaly Detection), reliability, scalability, single point of failure, safety, flexibility.

Description of control systems

We have seen that in installations spanning over medium to large areas, there is a tendency to adopt architectures of distributed systems. In these cases, control processes are implemented on physical computers located near (or even inside) the devices where low control latency is required, and on remote machines - both physical and virtual - where communication channels and the network ensure the necessary bandwidth and latency.
We have seen that the architecture of a control system that needs to operate over a long time horizon should be as independent as possible from any specific hardware and/or from those currently considered *reliable* standards, since they are all destined for obsolescence.

Keywords: acquisition bus, VME, cPCI, µTCA (microTCA), field bus, ethernet, embedded computers, virtual machines, machine learning, machine protection, AI.

Data path and data retention

In a system composed of many control processes on one side (e.g., the area of a particle accelerator) and many operator stations on the other side (e.g., a control room), there are numerous ways to organize data traffic and command forwarding.
In particular, the difference between the method based on queries (from consoles to control processes) and that based on the use of a centralized *cache* (a volatile memory area where control processes and consoles can write to and read from) is significant. We have also described the difference between *relational* and *non-relational* databases.
In addition to these architectural aspects, we also talked about how important it is to go through a *abstraction* process aimed at standardizing families of different devices by brand and/or model but that serve the same function. At the end of this process, the system will operate only with *virtual* representations of objects (classes) that will have uniform sets of characteristics (properties) and possible actions (methods).
While discussing how data can be represented, we saw how *serialization* techniques (e.g., JSON) allow representing any set of data through a sequence of bytes (hence serialization) that, in addition to containing the data value, also describes its format.
Finally, we have pointed out the difference between relational and non-relational databases and how the latter are particularly suitable for holding serialized data.

Keywords: Relational Database, memcached[1], object DB, key-value DB, data serialization, JSON, BSON, XML, YAML

User interface

[1] https://memcached.org/

The interaction of users (machine operators, machine physicists, external users, etc.) with the Control System should be as intuitive as possible and updated compared to the web and smartphones interfaces that we use daily.
Strangely enough, this is one of the most challenging aspects in the realization of a control system since producing an ergonomic and modern human-machine interface is extremely time-consuming. On the other hand, this field is rapidly evolving (outside the scientific community) and very attractive and functional graphical data presentation environments are starting to become available (e.g., open-source software like Grafana[2] is widely used at LNF).

Keywords: WEB interface, GUI.

Practical test

For the laboratory activity, we worked on individual workstations consisting of Intel® NUC Mini PCs with quad-core i5 processors, 500 GB SSD, 16 GB RAM, and 27" monitors with 2K resolution. The workstations were networked via WiFi with a bitrate ≥ 100 Mbps, sufficient for the planned activities. The programming environment used was LabVIEW[3] v 2023 Q1 professional edition.
We used a main program template based on the use of pre-existing events.

The experience involved accessing an image captured by the synchrotron light monitor of the DAFNE electron ring and processing it. The image was continuously updated at a frequency of ~2Hz in the real time cache of the DANTE Control System.
(The description of the experience is not provided.).

Keywords: false colors, memcached.

Alessandro Stecchi e Giampiero Di Pirro
(Tutors of the "Controls" session)

---

[2] https://grafana.com/

[3] https://www.ni.com/it-it/shop/labview.html