A parallel future

Continuing innovation for next-generation real-time controllers through software

MICHAEL WAHLER, SASCHA STOETER, MANUEL ORIOL, MARTIN NAEDELE, ATUL KUMAR – Programmable logic controllers (PLCs) started out as mere replacements for banks of hard-wired relays. Today's evolved digital controllers play an integral role in automation and power systems. Recent innovations in CPU and network technology have opened up new opportunities for parallelized future control systems, such as software-based fault tolerance and real-time data analysis. Before these benefits can be unlocked, however, many challenges must be overcome. ABB Corporate Research is actively pursuing the future of real-time controllers by tackling determinism, flexibility, scalability and sequential algorithms.

1 Acronyms used throughout the article

Acronym	Description
COTS	Commercial off-the-shelf
CPU	Central processing unit
DCS	Distributed control system
HMI	Human-machine interface
ICS	Industrial control system
MPC	Model-predictive control
PAC	Programmable automation controller
PID	Proportional–integral–derivative
PLC	Programmable logic controller
RTU	Remote terminal unit
SCADA	Supervisory control and data acquisition

The programmable logic controller (PLC) replaced hard-wired control relays ➔ 1. Today's PLCs have expanded well beyond their original design scope; they have to be much more than the relay logic replacement that they were initially designed to be. Born out of increased product diversity and innovation cycles, the PLC continues to evolve as new technologies are added to its capability. The result has been enhanced flexibility and reliability. By directly controlling equipment and processes, PLCs form the basis for supervisory control and data acquisition (SCADA), distributed control system (DCS) installations and programmable automation controllers (PACs). These industrial control systems have been around for decades. There are several flavors of control systems that each evolved due to very specific business needs and technological capabilities at the time of introduction.

Larger manufacturing processes with many sub-components are served by a DCS. The complete system benefits from the exchange of data that can be transmitted digitally among its nodes. This communication advantage allows, among other things, for a powerful human-machine interface (HMI). As commercial off-the-shelf (COTS) hardware is employed more and more, the focus of DCS shifts toward software differentiators including service. Rather than controlling processes like a DCS, SCADA takes a higher-level view. It is further removed from real-time constraints and coordinates processes. A PAC is a programmable microprocessor device that combines the functions of the PLC with greater flexibility in programming and tight integration with the other parts of the DCS. PACs are used for discrete manufacturing, process control and remote monitoring applications.

As technology evolves, especially the capabilities of hardware, networking, as well as the software systems, the boundaries become blurred. Moving forward, it is expected that the terminology will remain significant for historical reasons only, while the functionality will become unified. Many of these exciting developments will take place on the controllers themselves. They are the main and necessary interface between the process and the higher levels of the automation pyramid.

The current families of ABB controllers serve all automation areas well. For instance, some applications require fast cycle times, while others emphasize connectivity. ABB offers state-of-the-art products for each application.

The future of controllers

Performance gains have, in the past, come naturally with each new controller generation. An increase in central processing unit (CPU) frequencies was directly and positively reflected in application speeds. Unfortunately, this "free lunch is over" [1] because clock frequencies no longer increase as before. One reason is that faster frequencies result in additional heat that needs to be moved off the device. Alas, most controllers disallow moving parts that are needed for active cooling (eg, fans). As a result, lower clock speeds must be set than are supported by the CPU. Another reason is that the maximum clock speeds for many modern CPUs are declining. Instead, new CPUs provide several options for executing code in parallel. They are able to work on multiple data items simultaneously (eg, as is done by the MMX instruction set for Pentium and its many subsequent extensions) or are capable of processing on multiple cores, including hyper-threading. Performance gains for future controllers will come from exploiting these parallel mechanisms, while hiding the complexity from control application engineers ➔ 2.

Challenge: Scalability

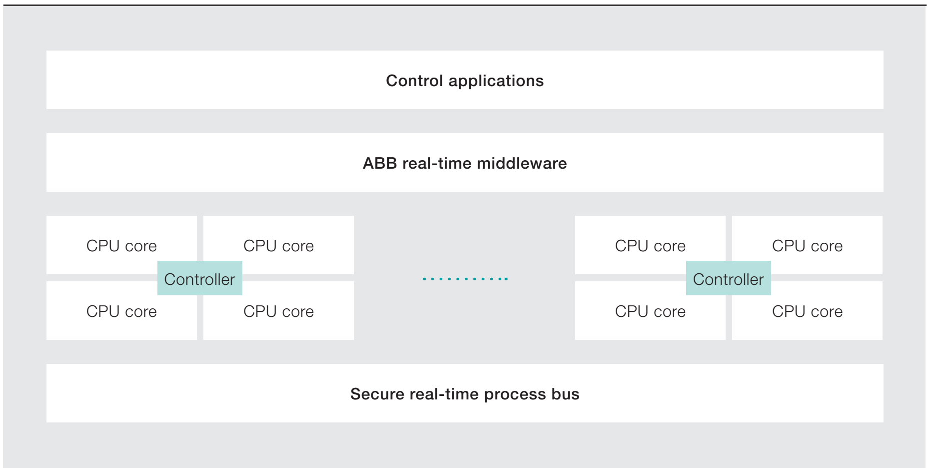
Distributed control applications are currently built in an ad hoc manner: with independent programs connected by a communication layer. Exploiting dual-core or quad-core CPUs is relatively easy; it can be achieved manually by either making small changes to the software or simply by letting operating systems distribute processes over the cores if they are ready to execute and can be executed in parallel. With the pervasiveness of multi-core chips, future generations of controllers will include a variable number of cores and hosts, making such a static allocation approach obsolete. The challenge is scaling software for an arbitrary number of computation resources, as a 128-core CPU or a 1024-CPU distributed system are an expected reality in a few years' time.

One solution to the scalability challenge is to break monolithic control applications into smaller components. The components are then scheduled to statically execute on the available computation resources (cores and hosts) [2]. Depending on the given deployment, a middleware chooses a suitable communication protocol for each pair of connected components.

This solution is not far-fetched, as it already corresponds to the control engineer's view of an application, where ladder logic or function block diagrams are commonly used to craft control applications. These

Title picture
Complex control systems benefit from parallel software

2 Envisioned system architecture for distributed real-time controllers



diagrams demonstrate that mutually independent components can be executed in parallel ➔ 3. Tragically, previous CPU generations are restricted to executing programs strictly one step at a time. Therefore, current technology must transform inherently parallel diagrams into sequential code, which is difficult and causes computational overhead. In contrast, parallel execution can exploit parallel branches in the control diagrams and make systems smarter.

Challenge: Determinism

Determinism is the quality of a system to behave in a predictable manner. If a controller receives the same input, it is expected to react the same way each time. With multi-core processors this becomes hard to guarantee. The reasons are context switches, caching behavior, and task synchronization. State-of-the-art, real-time control systems execute multiple concurrent control applications using operating system mechanisms such as processes, mutexes, or message queues. Such mechanisms leave a high degree of freedom to developers but are often difficult to deal with: they incur runtime overhead (eg, context switches between threads) and often require tedious and costly fine-tuning (eg, of process and thread priorities). Reuse is often made more difficult by the tight coupling of software to a given hardware or other software.

By employing a software architecture and execution framework for cyclic control applications, the construction of real-time control systems can be simplified, while increasing predictability and reducing runtime overhead and coupling [3].

Challenge: Flexibility

Speaking of determinism, with the limited computation power of single-core processors, reliably deterministic systems are often static – once configured and deployed they run until they reach the end of their lifetime. Changes can only be applied during maintenance windows, which may occur as rarely as once a year. This stands in strong contrast to real-world requirements as plants grow and evolve. Production becomes increasingly flexible up to mass customization approaches. At the same time, control algorithms will have to evolve during the lifetime of a controller.

> Performance gains for future controllers will come from exploiting parallel mechanisms, while hiding the complexity from control application engineers.

Patches against newly discovered security threats will constantly be issued and need to be integrated quickly into the control system. The research question is thus: Can we build control systems that are both flexible and deterministic? Early research results show that we can [4]. Using novel software architectures it will be possible to reconfigure control systems at runtime and adapt them to changing environments. Think controller migration with zero downtime. Think up-to-date software every day. Think control systems that easily adapt to your changing needs.

Challenge: Sequential algorithms

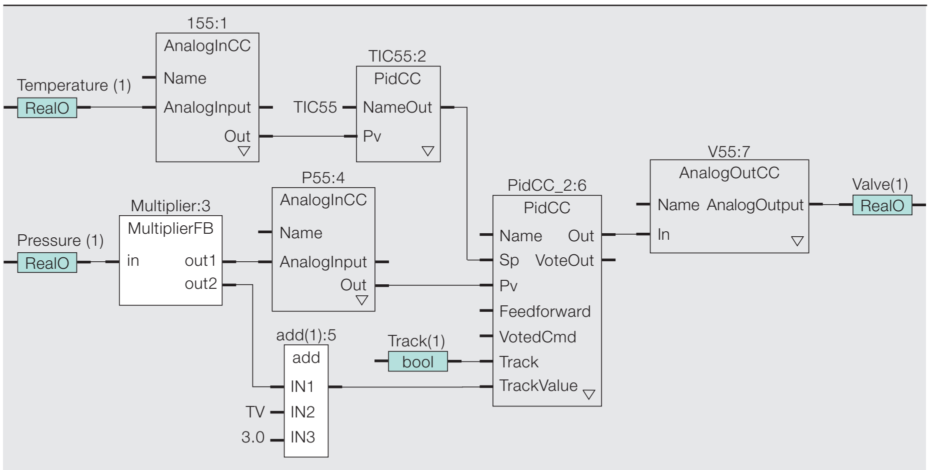
Control algorithms such as PID (proportional–integral–derivative) or MPC (model-predictive control) are described by mathematical formulas. In the past decades, researchers have devoted significant effort into developing and optimizing implementations for these algorithms. Thanks to these efforts, even computationally complex algorithms such as MPC can be executed at a microsecond time scale in today's systems.

Further improvements to control algorithms will necessarily come through parallelization. The challenges are clear: what parts of the algorithms that have always been executed in a sequential way can be executed in parallel? How can we optimize data access for parallel branches of the code? ABB control theory scientists are collaborating intensively with ABB computer scientists to provide high-performance parallel implementations of control algorithms [5].

Opportunities

The good news is that the solutions to these challenges give rise to new opportunities for future controllers:
– Hardware consolidation allows one to combine functionality on the same piece of hardware. Current practice is to use dedicated controllers for specific functional requirements.
  Using the increased flexibility offered by future controllers it will become possible

3 Example control application with parallel branches



By employing a software architecture and execution framework for cyclic control applications, the construction of real-time control systems can be simplified, while increasing predictability and reducing runtime overhead and coupling.

to place broad sets of functionality on the same physical device. This development will reduce the cost of control systems.

– Multi-core processors can provide more computation power than current single-core processors at significantly lower clock rates. They are easier to cool and thus, high-performance controllers will fit into small form factors.
– Integrated safety and software-based fault tolerance also contribute to hardware consolidation by using hypervisor and related isolation technologies offered by state-of-the-art CPUs. Using smart deployments, ABB scientists are working on achieving equal or even higher levels of safety and fault tolerance with the same or less amounts of hardware.
– High levels of cyber security can be supported by using some of the multi-core processing power for signing and/or encrypting network communication or other concurrent security checking activities. This is enabling technology for advanced features such as remote operations or DCSs. The importance of cyber security for industrial control systems is highlighted in another article in this issue of ABB Review [6].
– Additional functionality can be provided by the unlocked performance of parallel controllers. For instance, instead of time-consuming offline data analysis, data can be analyzed in real-time while control algorithms are executed. This will help optimize process efficiency as well as equipment longevity due to improved monitoring for predictive maintenance. Additional computation power offered by multi-core CPUs can also be used to increase the accuracy of control algorithms (eg, model-predictive control).

Summary

Software is the key to unlocking the inherent performance promise of multi-core CPUs. By moving away from hardware-centric evolution, it is possible to overcome future bottlenecks and to build controllers that scale with the ever-increasing demands of our customer's applications.

Michael Wahler
Sascha Stoeter
Manuel Oriol
Martin Naedele
ABB Corporate Research
Dättwil, Switzerland
michael.wahler@ch.abb.com
sascha.stoeter@ch.abb.com
manuel.oriol@ch.abb.com
martin.naedele@ch.abb.com

Atul Kumar
ABB Corporate Research
Bangalore, India
atulkumar.d@in.abb.com

References

[1] Sutter, H. (2005). A fundamental turn toward concurrency in software. *Dr. Dobb's Journal 30* Retrieved May 22, 2012 from http://www.drdobbs.com/architecture-and-design/184405990.
[2] Oriol, M., Steiger, M., Stoeter, S., Vardar, E., Wahler, M., Koziolek, H., Kumar, A. (2012, June) *FASA: A Scalable Software Framework for Distributed Control Systems.* International Symposium on Architecting Critical Systems (ISARCS). Bertinoro, Italy.
[3] Richter, S., Wahler, M., Kumar. A. (2011) *A Framework for Component-Based Real-Time Control Applications.* 13th Real-Time Linux Workshop, Prague, Czech Republic.
[4] Wahler, M., Richter, S., Kumar, S., Oriol, M. (2011) Non-disruptive Large-scale Component Updates for Real-time Controllers. In *Workshops Proceedings of the 27th International Conference on Data Engineering, ICDE,* Hannover, Germany (174–178).
[5] Li, Z., Donde, V., Tournier, J., Yang. F. (2011, July) *On Limitations of Traditional Multi-core Architecture and Potential of Many-Core Processing Architectures for Sparse Linear Solvers used in Large Scale Power System Applications.* Paper presented at the meeting of the IEEE Power Energy Society General Meeting. Detroit, MI, USA.
[6] Obermeier, S., Stoeter, S., Schierholz, R., Braendle. M. Cyber security: Protecting critical infrastructure in a changing world. *ABB Review 3/2012.*