Building safe, secure, connected cobots

By Ian Ferguson, Vice President, Marketing and Strategic Alliances, Lynx Software Technologies

Factories are increasingly expected to be nimble and adept. Instead of creating hundreds, thousands or even millions of the same thing, the new move is toward flexibility, allowing swift reconfiguration of equipment to support different builds. This might involve information that is extremely sensitive, which could result in OEMs writing some or all of the associated manufacturing software themselves. For the company whose robot is building a specific product, it needs to be able to integrate the OEM's software and ensure it doesn't interfere with the robot's underlying control software.

Facing a real opportunity to revolutionise certain factory workflows and generate new 'as a service' revenue streams, robotics is at an extremely exciting point right now. The traditional industrial robotics sector featuring companies like Mitsubishi, ABB Robotics, Omron and Fanuc needs to adapt to the new trend, as it will face both new opportunities and new competition, addressing diverse additional use cases that demand more mobility and new sensor technologies and price points. We are seeing a new level of dexterity in the way robots handle delicate or oddly-shaped objects. Emerging robots will have consolidated designs with multiple features sharing a heterogeneous multicore chip, and be connected.

Facing a real opportunity to revolutionise certain factory workflows and generate new 'as a service' revenue streams, robotics is at an extremely exciting point right now

Cobots to the rescue

The emergence of collaborative robots, or cobots for short, is all about interaction and joint work with people in the workplace. Cobots stand in stark contrast to traditional industrial robots which work autonomously, with operator safety guaranteed through isolation. Cobots currently are a very small percentage of the overall robotics market, but are an area expected to grow rapidly in the next five years, especially in manufacturing, healthcare and retail.

The Covid pandemic has accelerated this trend, leading to more acceptance of increased digital transformation and automation. Environments where six feet of separation between people are challenging will make good use of this technology. Large companies

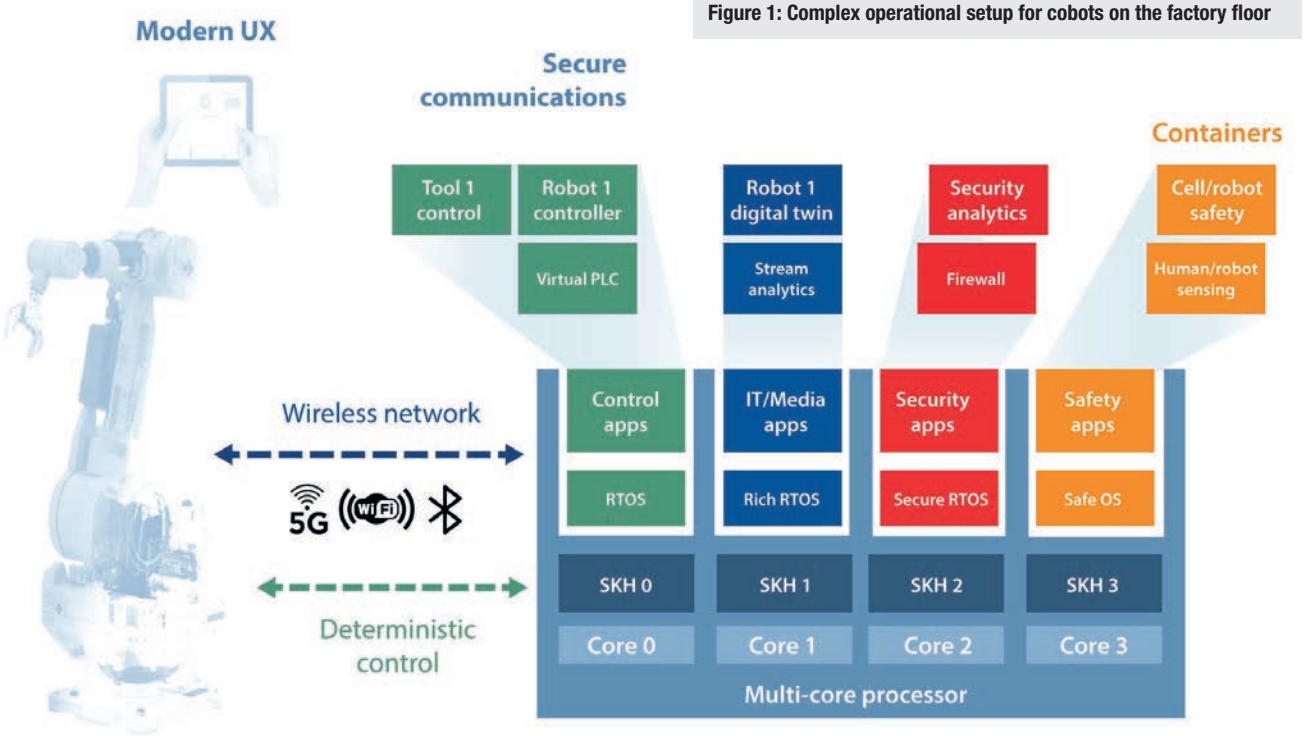

Figure 1: Complex operational setup for cobots on the factory floor

like Walmart are already using robots to clean their stores, with the potential for oversized roombas to clean schools, business premises and hospitals.

As cobots work in proximity to human staff, it's important that the they work all the time in a deterministic way to guarantee operator safety. Safety certifications and processes need to be defined and strictly observed if this market is to form successfully. This is an area where a lot of focus is on AI to improve cooperation with these machines. It also means there must be a secure way to deliver software updates to add new capabilities to these platforms.

For some of the larger platforms, there are moves toward greater modularity, enabling hardware upgrades over time. Manufacturers don't want to be tied to a specific hardware vendor, especially in an area as dynamic as artificial intelligence, where in the next few years there might be many business changes, such as acquisitions, companies ceasing to trade, and changes to performance leadership rankings.

IT/OT consolidation

Key consideration in cobot design is the consolidation of IT with operational technology (OT), blending robot instructions with factory operation. In a factory the rate of change is relatively slow, since the focus is on reliability, which translates to increased uptime, reduced accidents, less scrap, and so on. OT is therefore separate networks that get connected to IT networks at a main console level. The desire is to push this IT/OT fusion out onto the factory floor, so that machines can make more informed decisions more quickly. This challenge has to be addressed in the context of cost, power and footprint to make the cobot proposition commercially attractive. This implies shrinking multiple systems onto a single consolidated board – and, increasingly, a single heterogeneous multicore chip. These systems need to run rich operating systems like Linux and Windows, whilst also guaranteeing the platform's stable operation; they are referred to as mixed-criticality systems. Applications must be compartmentalised to ensure that some of them don't fail any part of the system.

Real-time systems based on a single core processor (SCP) are well understood in industry, which has adopted real-time system engineering processes built on the assumption of constant worst-case execution time (WCET). This states that the measured worst-case execution time of a software task when executed alone on a single core is the same when that task runs together with other tasks. This fundamental assumption is the foundation for the schedulability analysis, determining that a scheduling sequence can be found to complete all the tasks within deadlines.

In industry, the technological trend toward adopting multicore processor (MCP) systems is already well established. Many of these multicore systems have been designed with the speed and efficiency requirements of IT applications in mind, and do not always respect the predictability requirements of control systems in avionics, automotive, industrial automation, and others. In fact, whilst the assumption of a constant WCET is correct for single-core chips, it is not true for current multicore chips, due to interference across cores in accessing shared resources. Interference

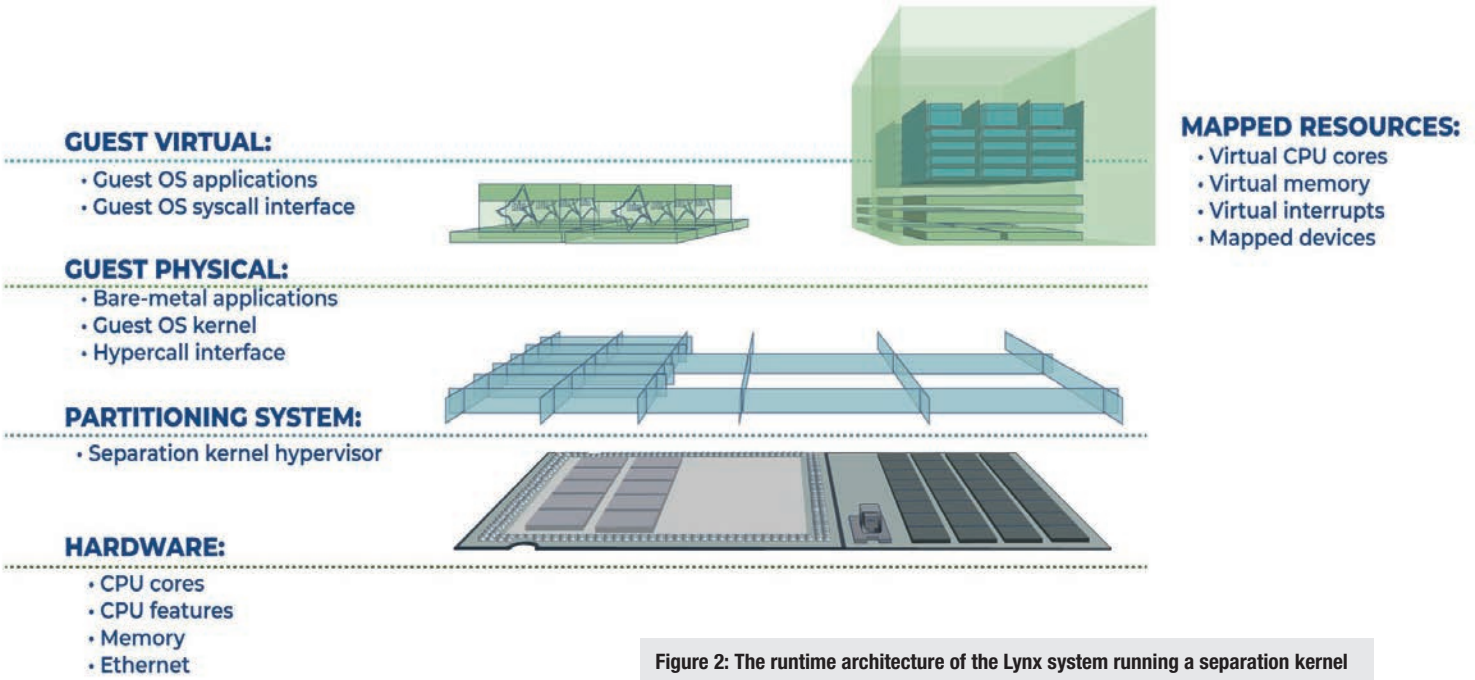

Figure 2: The runtime architecture of the Lynx system running a separation kernel

causes spikes in worst-case execution times more than six-fold compared to a single core. This is where Lynx focuses, providing a secure hypervisor that can securely partition and isolate applications, whilst guaranteeing that there is microsecond-level response to time-critical events.

Connected

These issues are further compounded by the fact that cobots need to be connected, for sharing sensor data, receiving instructions, software updates and their monitoring. The security aspect will impact the system and any device accessible from that network connection. To avoid this, security must be incorporated from the beginning as opposed to retrofitted. There must be an early alert when the system has been compromised, coupled with a way to bring it back to a known good state. And since this is not a static environment, being able to securely download new software to keep the system secure is a must. From a software perspective, the system must be built so it can't be reconfigured after boot-up and so that no application can accidentally or otherwise cause the robot to fail.

The emergence of collaborative robots is all about interaction and joint work with people in the workplace, in contrast to traditional industrial robots which work autonomously

A Lynx system running a separation kernel can meet these requirements; see Figure 2. In most deployments, there will be two platform kernels running:

1. The Separation Kernel controlling the physical hardware;
2. The OS kernel hosting guest applications.

This introduces an extra abstraction layer, compared with typical OS designs where hardware control is integrated with the OS kernel. Here, the separation kernel is the only piece of software with access to physical hardware; the application kernels have no access to real hardware and can only manage "mapped resources".

For this "software-assisted hardware partitioning model", all application partition boundaries are exclusively hardware-enforced according to a model. Here, if an application violates a partition boundary, hardware first catches the violation and requests software assistance from the Separation Kernel Hypervisor to manage the exception.

Computing

Cobots are about edge computing, which relates to shifting the intelligence nearer to where data is gathered. For latency, network availability and privacy reasons, these cobots will usually harness local on-premise computing resources instead of accessing the cloud. To do so successfully, in a context where they operate alongside and around people and undertake tasks with high safety risks, implies that their design must be safe and secure. EW