TRANSFORMA INSIGHTS

Sponsored By

doppelio

The Strategic Case for IoT Virtualization in Connected Product Engineering

The Strategic Case for IoT Virtualization in Connected Product Engineering

The 'left-shift' trend in IoT development emphasises earlier testing in the product lifecycle to reduce costs, accelerate development timelines, test timelines, and improve quality. This shift is driven by virtualised environments and simulation tools that enable testing without relying on physical devices.

Virtual testing allows developers to simulate diverse deployment scenarios, test at scale, and run cloud, edge, and device software development in parallel. Virtualised test environments are particularly beneficial for large scale deployments, complex IoT applications and mission critical connected systems, including AI-enabled devices (AIoT). They can also support continuous integration/deployment. A robust virtual test platform should be flexible to support fault injection and test automation, benefiting multiple stakeholders including OEMs, solution providers, service operators, integrators, and end-users.

Introducing virtualised IoT simulation for software testing

'Left-shift' is an increasingly common theme in development and testing for products and integrated systems. The term is intended to encapsulate an emerging trend to undertake an ever-increasing share of testing for products and services earlier in the development cycle, seeking to accelerate development and test cycles, to reduce the costs of solution development and deliver higher quality results.

Today's left-shift dynamic is enabled by ever more powerful emulation software. These tools are increasingly critical as products, solutions and turnkey services are becoming more complex leading to a significant uptick in the complexity of associated testing. In particular, products and solutions will often incorporate IoT-enabled devices, relying on application functionality that is distributed across multiple domains (including on-device, edge, cloud, and enterprise application integration). This results in much more complex test processes across multiple domains and including testing the interconnections between each domain.

One of the main challenges for left-shifting IoT testing is that IoT solutions are inherently associated with physical devices, ranging from simple smart meters through to sophisticated CCTV cameras with on-board AI analytics to connected vehicles with sophisticated software both on- and off-board. Due to the challenges of appropriately emulating IoT devices in software environments, the traditional approach to testing IoT solutions has been to perform bench tests or field tests with physical devices. Critically, this approach depends on the availability of physical devices. As such it is limited by the timeframes in which devices can be available, plus the timeframes associated with iterating hardware designs. Accordingly, overreliance on field testing is not only expensive but exposes a business to risks of extended release timelines if issues surface later in a development cycle.

Meanwhile, software is increasingly driving product and solution differentiation and customer value. The potential cost of software failures is therefore increasingly significant. The likely business impact, economic and reputational consequences of failures in the marketplace are increasing in-step. Consequently, there is an emerging need for high coverage testing to mitigate, as much as possible, the risk of failure in live field environments.

Accordingly, overreliance on field testing is not only expensive but exposes a business to risks of extended release timelines if issues surface later in a development cycle

The benefits of virtualised IoT simulation for software testing

Virtualised simulation of IoT devices is not dependent on physical devices and results in the ability to test both functional and non-functional (for example, connectivity network performance) aspects in a virtual environment. Applying such techniques can result in many potential problems or solution refinements being identified at an earlier stage in the development cycle resulting in a more agile IoT development cycle overall. Without virtual test techniques the risk of finding problems during field testing increases, which can result in comparatively extended development cycles and higher costs. Meanwhile, if required solution refinements are identified only once solutions have been deployed into live environments this can result in significantly higher costs to make the necessary refinements, including potentially field visits, product recalls and brand erosion. Such testing is not a one-time activity. Connected product applications can undergo multiple release cycles in a year requiring each release to be tested for functional and non-functional requirements.

Additionally, physical test environments for IoT devices are more limited in terms of the potential to test solutions in specific deployment scenarios or to test for unusual 'edge cases'. This is particularly reelvant when tests would ideally involve the interaction of multiple IoT devices, or multiple IoT device types, within a wider system or solution. Physical testing at scale is usually infeasible before full deployment, especially for systems that might include tens of thousands or even millions of connected assets. Physical test environments are similarly challenged in terms of ability to address load testing.

Virtualised environments can be particularly valuable when a product OEM or solution provider intends to ship cellular connected devices to multiple countries, enabling specific network connectivity environments to be emulated in virtual environments, so providing insight into how a device will behave in an extensive range of network conditions.

Clearly, physical deployment environments can be extremely complex and the closer that a virtualised test environment can be to accurately mirroring physical environments, then the more effective and valuable will that virtual test environment be.

Emulating cloud and edge for virtualised IoT software testing

IoT solutions are particularly complex to test fully in virtual environments, since these are inherently connected devices with aspects of solution functionality deployed on-board devices and in remote (often cloud) locations. Accordingly, to fully test IoT software solutions, it is necessary to test both on-device and cloud functionality, and also interactions between these two environments.

IoT applications and connected assets are also becoming more complex. For example, today's connected cars benefit from distributed services with multiple applications on-board, many of which must handle high volumes of data (including time-critical data, highly detailed information, video and LiDAR streams and more besides). Accordingly, today's connected vehicle environments are characterised by the presence of ever more data sources and data consumers on device, plus more complex interfaces with cloud applications, making rigorous testing both more complex and more critical. Meanwhile, on-vehicle software can be frequently updated resulting in a need for regular ongoing testing. Conversely, a more mundane application like smart metering may seem relatively simple but an overall smart metering system will often be a very complex solution due to the huge numbers of meters typically deployed as part of a solution.

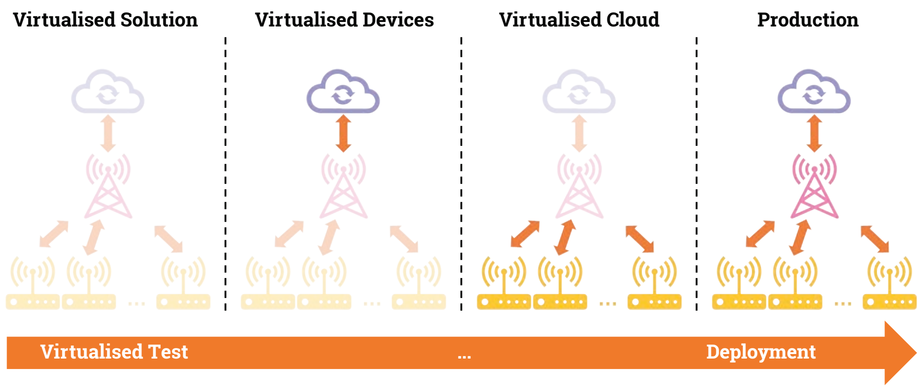
This highlights a significant advantage of virtualised testing for IoT solutions, which is that on-device and cloud environments can be tested separately. For example, in a virtual test environment the testing of a cloud application is no longer dependent on the availability of test hardware; virtualised IoT devices can simply plug-in to cloud applications to emulate the behaviour of field devices. And, at an appropriate point in the test cycle, physical test devices can similarly connect to emulated cloud environments for testing.

Meanwhile, software onboard IoT enabled assets can be tested in a virtualised environment, including applications and any supporting operating systems, all benefitting from emulated data feeds from virtualised sensors together with emulated interactions with cloud applications.

Effectively, this allows the development of device hardware, on-device software, and cloud applications to proceed in parallel, with each element benefitting from interfaces to emulated components until relatively late in the development cycle. In particular, the availability of physical test devices becomes significantly less critical to the overall development cycle, since many aspects of development and test can be undertaken without the need for a physical device. Early integration and end-to-end system testing with virtualisation enables faster testing, testing for corner cases, and automated testing.

Additionally, virtualised test environments allow for testing at a scale that would not be feasible in a physical test environment. For example, it is relatively simple for a virtualised test environment to support testing of a full smart meter rollout, potentially including millions of devices, or a more complex integrated system such as a warehouse management system.

Virtualised environments are also well-suited to support the testing of new software versions, including on-board application software, cloud application software and any relevant operating systems as these evolve over time.



A platform for virtualised IoT software testing

As a general rule, a virtualised IoT software test environment should support both simulated and emulated twins that are essentially indistinguishable from real devices and environments

Platform environments are a natural fit for virtualised IoT software testing. Whilst it would be possible to create customised simulators to support any testing requirements, there are many aspects of software testing that will be common to multiple solution types and so which would be best supported by a platform environment due to benefits of scale.

For example, the potential to test multiple devices, or an entire device estate, is a functionality that a virtualised IoT test platform should make available with a few clicks, whereas this would likely be a significant development initiative for a customised test solution.

As a general rule, a virtualised IoT software test environment should support both simulated and emulated twins that are essentially indistinguishable from real devices and environments. It should also support a wide range of communications protocols including, for example, Message Queuing Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), Constrained Application Protocol (CoAP), BACnet, and also domain specific standards such as HL7 (Health Level 7) for exchanging electronic health information between different healthcare systems.

A virtualised test platform should also support a wide range of connectivity network types and be able to generate associated simulations. These include wide area connectivity types (for example, cellular or satellite connectivity) and also specific aspects of local networking including, for example, multicasting, IP spoofing, routable virtual IPs, Virtual Router Redundancy Protocol (VRRP) and support for IPv6 and IPv4 where such functionality will be a characteristic of anticipated deployment environments.

Virtualised IoT test environments should also be well suited to supporting distributed development teams, so that, for example, a cloud application development team in one geographic location can work together effectively with a device and device software development teams located in different geographic locations.

A platform should also support testing of abnormal behaviours and fault injection, to ensure that all aspects of an overall solution are fully understood in the case of malfunctions or connectivity outages.

Additionally, platform environments should support the development and automation of test programmes, to ease the test and release of future software versions or hardware variants and upgrades. Such platforms should also leverage generative AI to support interfaces to software engineers and test personnel, for instance by offering a natural language interface that engineers can use to configure test cases or to support troubleshooting. Agentic AI techniques should also be used to support a range of test and configuration procedures.

Ultimately, a virtualised platform for IoT testing should be sufficiently flexible to reflect the full complexity of a wide range IoT solutions, while resulting in a lower cost to achieve a certain level of quality.

Implementing virtual IoT software testing

For potential adopters of virtual IoT software testing, the process should be smooth. Since software-based device simulators are intended to emulate IoT devices as accurately as possible, integrating these into an already existing environment should simply be a matter of connecting the 'outputs' of an emulated IoT device to the 'inputs' of a cloud application. Any existing test programmes should easily migrate from hardware-based to software-based.

Any such adopter would also benefit from the potential for parallel development of cloud and edge applications, both with emulated interfaces to a virtual environment. Additionally, such virtual testing environments can better support testing across multiple development projects that might be in-flight at the same time.

Effectively, the adoption of a virtualised test environment for IoT devices allows IoT solutions to be integrated into a continuous integration and continuous deployment (CI/CD) workflow.

The impact of AIoT

As the costs and power-consumption of on-device processing fall, more devices will be enabled with on-board AI capabilities

AI is increasingly being deployed on-board IoT devices in the form of AIoT. For example, many CCTV cameras now have on-board AI capabilities that support image analysis to identify specific behaviours within a crowd, or to identify individuals, or simply to read car licence plate information. As the costs and power-consumption of on-device processing fall, more devices will be enabled with on-board AI capabilities.

This will have a significant impact for many estates of IoT devices, since AI solutions can be frequently updated and enhanced. Effectively, AIoT results in a requirement for CI/CD for devices in the field, but with much faster iterations.

Additionally, since AI algorithms can behave differently when hosted in varying hardware and operating system environments, the effectiveness of any updated or refined AI algorithms will need to be tested for the blend of hardware types and software environments that may be in place in the field as part of an existing device estate. The testing should draw on both real and synthetic data, to identify how different hardware, operating system, and application environments perform when faced with the full diversity of inputs that will be encountered during field operations. Testing should also encompass edge cases, latency scenarios, and seek to identify performance degradation across different environments.

Effectively, the dynamics of AIoT mean that the virtualised IoT test environments that would be beneficial for the development and deployment of IoT solutions, as discussed above, can become a critical aspect of the operations support infrastructure for AIoT.

Who can benefit from virtualised IoT test environments

A wide range of stakeholders can benefit from virtualised IoT testing. Firstly, connected device OEMs can benefit from the potential to rigorously test solutions in a wide variety of contexts, so ensuring that only fully robust and well-tested solutions are released to market. OEMs can also benefit from accelerated development cycles and the potential to more efficiently customise or enhance solutions according to customer needs.

However, a wide variety of stakeholders beyond OEMs can also benefit, including:

- Solution providers, who will be able to more efficiently refine and optimise the solutions that they develop.
- Communications service providers, particularly cellular and satellite operators, who will be able to rigorously test IoT solutions before they commit to support those solutions.
- Systems integrators, who will be able to approach project deployment phases with lowered risk, confident that the solutions that they have developed for a client will work effectively in real world environments.

And, of course, end-users and adopters of IoT solutions will also benefit from the efficient development of high-quality solutions.

OEMs can also benefit from accelerated development cycles and the potential to more efficiently customise or enhance solutions according to customer needs

Conclusions

Virtualised IoT simulation enables earlier, faster, more cost-effective software testing by eliminating reliance on physical devices. It allows scalable, parallel development and testing across cloud, edge, and on-device systems, improving quality, reducing time-to-market, and supporting CI/CD workflows. It also facilitates testing of AIoT scenarios, edge cases, and diverse environments, benefiting OEMs, integrators, service providers, and end-users alike.

"Although we possess the expertise and knowledge to develop our own IoT simulation tools, we realised that doing so would shift focus from our core business objectives. With the increasing complexity of connected environments, we opted for a dedicated connected product test automation platform that allowed us to efficiently scale testing, accelerate time-to-market, and keep our teams laser-focused on innovation and execution."

Sr. Manager, Cloud and IoT Architecture, Rheem Manufacturing.

About Doppelio

In the complex world of IoT, with complex algorithms, infinite scenarios, a multitude of devices, and siloed teams, comprehensive testing is easier said than done.

Doppelio lets you Virtualize physical devices and Simulate their data behavior very easily. These simulated devices or Doppels and the test scenarios will help you test your IoT Applications comprehensively without dependence on physical devices. You can model not just individual devices but the entire system of things your application is required to work with. You can also inject real-world conditions such as network errors, and sensor faults, to test boundary conditions fully.

Compared to typical API or web testing tools – which were never designed for IoT testing – Doppelio virtualizes your physical devices including their state, their behavior over time, and their protocols, enabling you to achieve high fidelity with your actual physical devices.

A fully managed SaaS platform, Doppelio provides on-demand scaling which lets you test for scale and performance with ease, any time. Doppelio is designed as a low code platform, so your teams will become productive very quickly.

Doppelio is a single platform for multiple types of testing – Functional, Performance, Scale, Network, and Regression - and production systems monitoring for degradation of functionality or performance. You can also use it for Post Release Troubleshooting. It integrates with your appium and selenium scripts for mobile and web app testing to include end to end scenarios, via our rich API.