

Report on the EC Workshop on Virtualisation

September 29, 2009, Brussels.

Participants

Christian Bertin, STMicroelectronics
Attila Bilgic, KROHNE
Albert Cohen, INRIA
Koen De Bosschere, Ghent University
Bjorn De Sutter, Ghent University
Lieven Eeckhout, Ghent University
Elizabeth Gonzalez, Ruhr-University Bochum
John Goodacre, ARM
Steve Hand, Citrix and U. Cambridge
Jonas Maebe, Ghent University
Bilha Mendelson, IBM
Alasdair Rawsthorne, The University of Manchester
Daniel Scheibli, SAP
Stefaan Sonck Thiebaut, OpenSynergy
Zulema Olivan Tomas, European Commission
Panagiotis Tsarchopoulos, European Commission
Seppo Turunen, Nokia

1. Motivation

The previous Call for Proposals in the *Computing Systems* work programme resulted in one coordination action and nine STREPs. In some of those projects virtualisation is clearly present as an enabling technology, but none of the projects focus on specific research into virtualisation technology.

The FP7-funded HiPEAC NoE does have a *Binary Translation & Virtualisation* research cluster, however, and virtualisation also plays a central role in its recently submitted HiPEAC Vision document.

This workshop brings together several people from academia and industry to brainstorm on specific topics in the domains of server virtualisation and of embedded virtualisation.

In particular, the questions posed to the participants of the meeting are:

- What are the research challenges in virtualisation?
- Is there a need for European funding in this area?
- If so, what is the best way to cover this topic in future calls?
 - How to integrate virtualisation in a call (as an potential aspect, as a requirement, or as a challenge in its own right)
 - Which type of instrument is best suited?
 - How do we describe the challenges?
 - Is it, with respect to virtualisation, better to have vertical or horizontal integration? Do we need to impose requirements in this regard?

2. Summary of the position statements

Virtualisation : Tool for Secure SW and Service Integration in Mobile Devices

Seppo Turunen, Nokia

A prevailing trend on the mobile market is a shift away from competing by means of hardware differentiation towards competing through user interfaces, software applications and network services. An open source virtualisation framework could reduce the porting efforts that are currently required, better exploit the underlying hardware, and assist with various kinds of isolation (security, IPR).

Research challenges:

- Combining real-time and general purpose computing on the same hardware.
- Securing virtual appliances and processes.
- Defining virtualisation framework interface semantics (security, performance, ...) in a way that is useful for end users, developers and system providers.
- Hypervisor support for (multimedia) accelerators.

Virtualisation across the ARM ecosystem

John Goodacre, ARM

There is a common view in industry that virtualisation is a Good Thing™, since it enables chip consolidation, easier integration, and more portability, manageability and isolation. Paravirtualisation is however too expensive, because it requires rewriting the bottom third of an OS, the porting effort must be repeated for new releases, and the reliability of the rewriting is not guaranteed. Furthermore, adapting an OS is not always possible, and it also creates security issues. Finally, also in the embedded world, hardware support for virtualisation is now emerging.

Research challenges:

- Research in hardware support is only interesting if it is at least 5-10 years out, as R&D for hardware until 2015 is already in progress. In the mid-term, software has to make sure that existing hardware (lacking some virtualisation features) still works though.
- Security and isolation in converged software domains (e-commerce/finance, confidentiality, open/closed software, interference).
- Real-time support.
- General abstracted interface to the virtualisation layer.

Perspectives on Virtualisation

Daniel Scheibli, SAP

SAP is a huge user of (server) virtualisation, and considers server virtualisation to be already an established enabling technology that is becoming ubiquitous. At the same time, SAP is also itself working on a cloud-based virtualisation platform to deploy its solutions. In this context, dynamic resource allocation is very important.

Research challenges:

- Workload modelling (QoS, SLA) and benchmarks (cpu-bound, I/O-bound).
- Standardisation: VM descriptions, (management) interfaces, ...
- Interoperability and transparent migration between different clouds.
- Integrated virtualisation designs (taking advantage of the same code executed in multiple concurrent VMs).

Virtualisation in Europe

Alasdair Rawsthorne, The University of Manchester

New virtualisation technologies (memory, network, disk, cpu, ...) are always adopted in three stages: first introduced as a transparent layer, then APIs are added for management and

monitoring, and finally APIs are provided for full integration in the software stack. Process virtualisation (e.g., Java) is currently in stage 3, while system virtualisation is still at stage “2-”.

Research challenges:

- Fully integrating system virtualisation into the software stack to ease the development of the upper software layers → application-aware checkpointing, reversible & automated debugging, propagate SLAs from SW to HW, safe sandboxing, continuous test-in-production, autonomous scenario planning.
- Using dynamic code generation to optimise virtual environments.
- Bringing existing virtualisation capabilities in server virtualisation to network edge (mobile/embedded) devices → design space changes in power, CPU+DSP, security/identity/micro-payments.

Performance Modelling of Virtualised and Consolidated Systems: Challenges and Opportunities

Lieven Eeckhout, Ghent University

The ability to model virtualised workloads is very useful in assisting with optimal resource allocation within virtual environments to reach QoS & SLA requirements and to optimise application/HW mapping for optimal performance/Watt. This requires a top-down approach, rather than a bottom-up approach.

Research challenges:

- Selecting appropriate benchmarks, and devising sound benchmarking methodologies for virtualised environments.
- Modelling virtualised workloads.
- Dealing with non-determinism in benchmarking/modelling.
- Simulating (micro)architecture plus system software.

Virtualisation at ST and ST-Ericsson

Christian Bertin, STMicroelectronics

Virtualisation has three main uses within ST/ST-Ericsson: virtual platforms that enable evaluating systems before hardware is available, evaluating and analysing operating systems behaviour, and the use of process virtualisation (Java, JavaScript, Flash, .NET, ...). There is much interest in embedded OS virtualisation to enable easier driver reuse, to isolate real-time parts into their own layer, to ease software reuse, for open/closed software isolation, and for security isolation. Finally, traditional static compilation is unable to improve much further, while split compilation offers many extra opportunities.

Research challenges:

- Designing a common, generic and easy-to-use split compilation tool flow based on cross-development, cross-integration, cross-profiling, emulators and install-time compilation.
- Investigating optimal combination of optimisations to use in split compilation setups.

A Virtualisation Perspective to Regaining the Lost Performance Portability

Albert Cohen, INRIA

A lot of programmer productivity has been lost recently due to extra requirements put on the developers: porting, parallelisation, ... Reason: the hardware has become more flexible than the software (HiPEAC vision). Standardised bytecode-based virtualisation is required to regain this productivity and to enable deferring the analyses until runtime.

Research challenges:

- Defining an open and overarching bytecode-based virtualisation standard.
- Defining the appropriate metadata required to perform the required optimisations in the various backends for different kinds of hardware.

Virtualisation for Smart Cameras: from a vertical to a horizontal market

Bjorn De Sutter, Ghent University

Modern smart camera deployments and applications require distributed processing on the cameras themselves. These cameras consist of heterogeneous multi-core platforms, that are inflexible and which differ between suppliers and camera generations. As a result, programmer productivity is low, markets for specific software and hardware are small and vertical. A virtual bytecode platform would enable market horizontalisation and will increase programmer productivity.

Research challenges:

- One bytecode to rule them all: generic bytecode with metadata that enables the optimal exploitation of heterogeneous multi-cores using split/dynamic compilation.
- Performance modelling to enable VMs to choose the optimal HW for a particular application.
- Developing families of accelerators with split compilation/virtualisation in mind.

Virtualisation Trends on Mobile Embedded Systems

Elizabeth Gonzalez, Ruhr-University Bochum

System virtualisation is used on mobile devices to enable flexible resource usage, to combine real-time and non-real-time workloads on the same cpu, and for security and robustness isolation. Moreover, mobile platforms are heavy users of multi-core solutions to address multimedia, bandwidth and changing radio requirements. Software has to be able to take advantage of these HW capabilities.

Research challenges:

- Virtualisation for heterogeneous multi-cores.
- Adding HW virtualisation support to embedded systems.
- Load balancing to enable safely combining RT and non-RT systems.
- Power management in virtualised environments.

Embedded Virtualisation Challenges

Attila Bilgic, KROHNE

KROHNE produces flow meters, whereby components (HW & SW) have to be available for up to 20 years. A virtual SW platform would enable transparently switching hardware computing components, thereby reducing the time for which these must be available. A performance loss of 10-20% would even be acceptable, as long as real-time guarantees can be honoured.

Research challenges:

- Integrating real-time requirements in overall system design (missing a deadline every now and then can be acceptable, only if the system is designed to cope with this, which depends on the system but not on the individual components).
- Bring the transparent virtualisation functionality of desktop systems to embedded devices.
- A virtual platform that enables modular, reusable, hardware and platform-independent software.
- Support for functional safety (SIL = safety integration levels).
- Support for safe and secure remote updates in the field.

Virtualisation challenges at OpenSynergy

Stefaan Sonck Thiebaut, OpenSynergy

OpenSynergy develops embedded automotive software (for ECUs = engine control units). Currently, all ECUs are isolated, locally optimised systems connected via several busses (one bus per core functionality), with a gateway to interconnect the busses. Many cars have over

50 ECUs today though, so people are looking into consolidation and global optimisation. Virtualisation is indispensable in this case to guarantee isolation.

OpenSynergy currently uses a paravirtualisation approach on top of a micro-OS, but has some problems, such as driver usage across domains, and configuring the scheduler of the stack to satisfy RT requirements.

Research challenges:

- Very quick bootup procedures for a virtualised environment (first functionality available within 100 ms, full functionality after 5-10 secs).
- Efficiently sharing HW between different virtual domains, and having drivers for the HW under all virtualised systems.
- Defining scheduling policies for a set of virtual domains in terms that are understandable for both developers and users and the scheduler.
- Satisfying safety standards.
- Moving from paravirtualisation to full virtualisation.

Virtualisation challenges

Bilha Mendelson, IBM

There are three kinds of virtualisation: process virtualisation, OS/system virtualisation, and cloud-based virtualisation (on top of which in turn many process/system VMs can run). There are moreover different abstraction levels: the hypervisor, the OS, the library level and the application/programming language level.

Research challenges:

- Performance analysis and optimisation in virtualised environments: virtualisation hides all details traditionally used to analyse performance. Requires cross-layer collaboration between compiler, OS, VM(s), I/O, may need emulation.
- Defining a bytecode format and the required metadata to enable the required dynamic optimisations, including for accelerators (GPGPU, ...).
- Debugging/monitoring virtual environments.
- Scalability.
- Reducing complexity.
- Workload affinity models.
- Testing software that will run in virtual environments: it is unknown how the virtual resources will be mapped to physical resources at run time.

Research Challenges in Virtualisation

Steve Hand, Citrix and U. Cambridge

Virtualisation today is strongest in (private) data server virtualisation. More recently, client virtualisation (on the desktop, laptop, mobile devices) has also gained popularity. One particular form is Virtual Desktop Infrastructure (VDI), in which all of a user's data is contained within a single virtual environment. Another trend is cloud virtualisation: services providing VMs on-demand to customers (possibly including PaaS and SaaS).

Research challenges:

- Making virtualisation help more with green computing: take into account energy costs in cloud computing fees, locate data centers near remote sources of reusable energy → requires live migration across the world
- Personal VMs: VDIs that can be instantiated anywhere → global accessibility, privacy, durability, track provenance
- Trustworthy computing: make commercial/public cloud services acceptable to large corporations (rather than only private clouds): data security, auditing, isolation, information leakage.

3. Conclusions

What are the research challenges in virtualisation?

Concerning the challenges, it is clear from the position statements that there are some recurring themes in both server and embedded virtualisation, but that there are also major differences. It is also very clear that we are only at the very beginning of virtualisation of embedded systems, and that there are still many unanswered questions.

Recurring motivations for virtualisation are:

- Consolidation (of number of servers/number of processing elements)
- Isolation for security, confidentiality, IPR, combination of safety levels
- Management cost
- Portable performance

Technical challenges

- Full virtualisation solution for heterogeneous embedded platforms
- Design of virtualisation-ready hardware platforms (including virtualisation-aware accelerators)
- Definition of VM management interfaces and their standardisation
- Real-time guarantees for virtualised workloads, load balancing of RT and NRT tasks.
- Development of performance models, benchmarks, QoS, SLA
- IO-virtualisation – reuse of software/device drivers
- Development of a tool flow for the design of virtualised applications (split compilers, JIT compilers, profilers, support for feedback-directed, whole-program and workload-specific optimization in process virtualization)
- Full integration of virtualisation into software stacks.

Is there a need for European funding in this area?

For this question, it is important to make a distinction between server virtualisation and embedded virtualisation.

- **Server virtualisation** is a mature technology that has developed its own momentum and evolves very fast. A three year project with results that will be available five years from now is not useful. The major players in this area are already spending a lot of money on innovation.
- **Embedded virtualisation** is a completely different story. There are very few success stories so far and there are still numerous challenges to be tackled before this technology will become mainstream. Given its historical strength in embedded systems, it is important for Europe to take the lead in embedded virtualisation in different domains (mobile computing, automotive, air and space, industrial automation, ...). This effort might be crucial to consolidate Europe's leadership in the next generation embedded systems.

If so, what is the best way to cover this topic in future call?

- How to integrate virtualisation in a call (as a potential aspect, as a requirement, or as a challenge in its own right)
- Which type of instrument is best suited?
- How do we describe the challenges?
- Is it, with respect to virtualisation, better to have vertical or horizontal research? Do we need to impose requirements in this regard?

There seems to be a consensus that embedded virtualisation could benefit most from a large project (IP-style) developing a public embedded virtualisation platform that can be deployed across different embedded domains (mobile computing, automotive, air and space, industrial automation), and tackling the different challenges as identified during the meeting. This is a rationale for horizontal rather than for vertical integration. There are also good reasons to keep embedded virtualisation and data-center virtualisation solutions separated and to not impose that the solutions should be applicable in both areas as the technical challenges seem too far apart.

A working title for such a project could be:

Towards full virtualisation solutions for heterogeneous platforms including the design of virtualization-ready heterogeneous hardware platforms.