MIKELANGELO: MIcro KErneL virtualizAtioN for hiGh pErfOrmance cLOud and HPC Systems

Nico Struckmann[1(✉)], Yosandra Sandoval[1(✉)], Nadav Har'El[2], Fang Chen[3], Shiqing Fan[3], Justin Činkelj[4], Gregor Berginc[4], Peter Chronz[5], Niv Gilboa[6], Gabriel Scalosub[6], Kalman Meth[7], and John Kennedy[8]

[1] High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, Nobelstr. 19, 70569 Stuttgart, Germany
`{struckmann,sandoval}@hlrs.de`

[2] ScyllaDB, Ltd., 11 Galgalei Haplada, 4672211 Herzelia, Israel
`nyh@scylladb.com`

[3] European Research Center, Huawei Technologies Duesseldorf GmbH, Riesstr. 25, 80992 Munich, Germany
`{fang.chen1,shiqing.fan}@huawei.com`

[4] XLAB Razvoj programske opreme in svetovanje, d.o.o. (XLAB), Pot Za Brdom 100, 1000 Ljubljana, Slovenia
`{justin.cinkelj,gregor.berginc}@xlab.si`

[5] Gesellschaft fuer wissenschaftliche Datenverbeitung mbH Goettingen (GWDG), am Fassberg 11, 37077 Gottingen, Germany
`peter.chronz@gwdg.de`

[6] Ben-Gurion University of the Negev (BGU), Office of the President - Main Campus, 84105 Beer Sheva, Israel
`{gilboan,sgabriel}@bgu.ac.il`

[7] IBM Israel Science and Technology Ltd., 94 Derech em-Hamoshavot, 49527 Petach Tikva, Israel
`meth@il.ibm.com`

[8] Intel Shannon Limited (INTEL), Collinstown Industrial Park, Leixlip, Ireland
`john.m.kennedy@intel.com`

Abstract. MIKELANGELO is a project, targeted to disrupt the core underlying technologies of Cloud computing, enabling even bigger uptake of Cloud computing, HPC in the Cloud and Big Data technologies under one umbrella. The vision of it is to improve responsiveness, agility and security of the virtual infrastructure through packaged applications, using lean guest operating system OSv and superfast hypervisor SuperKVM. In short, the work will concentrate on improvement of virtual I/O in KVM, using additional virtio expertise, integrated with the light-weight operating system OSv and with enhanced Security. The HPC in the Cloud focus will be provided through involvement of a large HPC centre, with the ability and business need to cloudify their HPC business. The Consortium consists of hand-picked experts (e.g., the original creator of KVM - Avi Kivity) who participate in the overall effort to reduce one of the last performance hurdles in the virtualisation (I/O). Other layers of inefficiency are addressed through OSv (thin operating system). Such approach will allow for use of MIKELANGELO stack on heterogeneous infrastructures, with high responsiveness, agility and improved security. The targeted audience are primarily SMEs (e.g. simulation dependent SMEs) and data center operators who either benefit from higher performance or flexibility, introduced by the software stack. Four real world use-cases with clear owners, serve as validators and also directly contribute to the exploitation of project results.

Keywords: Cloud computing · HPC · Big data · SuperKVM · KVM Virtualised infrastructures · OSv · vTorque

1 Introduction

MIKELANGELO project[1] [1,3] is about improved performance, flexibility and manageability of virtualised infrastructures. The project deals with a wide range of technologies on many different levels of a typical stack, such as cloud and HPC. These involve the hypervisor, the guest operating system and the management layers. On top of that, this holistic approach puts MIKELANGELO into a unique position with the capacity of providing several cross-cutting technologies extending the potential of individual components. Today's Cloud and HPC architectures are inefficient, as there is always a trade-off between flexibility, efficiency, stability and security. Cloud environments are in general more flexible than static HPC environments, in terms of operating systems, kernel version, software availability and abstraction of compute environments, i.e. shared file system mount paths. However, Cloud's flexibility provided by virtualization comes with the disadvantage of a high overhead for I/O intensive applications. The overhead of I/O performance in virtual environments when compared to native performance of compute nodes is not suitable for many HPC workloads that are latency-sensitive or have high I/O rates [2]. The project addresses the whole stack comprising typical infrastructure as a service (IaaS) or, with the support of some of the use cases, even platform as a service (PaaS). Its core development focuses on enhancements made to the KVM hypervisor and several, both functional and non-functional, improvements of the guest operating system. The optimized hypervisor sKVM targets, in combination with the lean guest operating system and further optimizations for virtualized I/O, like virtual RDMA (vRDMA), improved I/O core scheduling (IOCM), Zero-Copy transmission (ZeCoRX) and shared memory between VMs running on the same host (UNCLOT), snap for holistic telemetry and SCAM for greater security, to improve the overall performance, flexibility and manageability of the two formerly distinct Cloud and HPC environments. All of these components are transparently integrated into commonly used middlewares, the widely used HPC batch-system PBS/Torque [4] for HPC environments and OpenStack for

[1] https://www.mikelangelo-project.eu.

Clouds. vTorque extends PBS/Torque and provides the baseline for the integration of several MIKELANGELO components into HPC environments. It comes with management capabilities by the creation of virtual environments and the deployment of batch workloads in such. While MCM enhance OpenStack by new scheduling capabilities, Scotty provides by continuous integration support. All components can also be deployed independently.

2 MIKELANGELO Architecture

This document describes the current status of the final architecture and provides an overview of all components involved and where they are located in the different layers the MIKELANGELO framework. The architecture is designed to improve the I/O performance in virtualised environments and also to bring the benefits of flexibility through virtualization to HPC systems. These benefits include, besides application packaging and application deployment, also elasticity during the application execution, without losing the high performance in computation and communication provided by HPC infrastructures. Each of the components can be used independently of other, however the benefits sum up combining as much components as possible. The diagram in Fig. 1 shows a high level overview of these components and their relations. The Hypervisor Architecture, sKVM, aims at improving performance and security at the lower layer of the architecture, the hypervisor. Previous work was divided into three separate components: IOcm, an optimization for virtual I/O devices that uses dedicated I/O cores, virtual RDMA (vRDMA) for low overhead communication between virtual machines, UNCLOT for shared memory between VMs running on the same host, and ZeCoRx avoids copying data between host and guest buffers on the receive path. Further SCAM, a security feature identifying and preventing cache side-channel attacks from malicious co-located virtual machines in multi tenant Cloud environments. The guest operating system (or “guest OS”) is the operating system, it implements the various APIs (Application Programming Interfaces) and ABIs (Application Binary Interfaces) which the applications utilize. The goals of enhancing the guest operating system are to run I/O-heavy and HPC (high performance computing) applications more efficiently than on traditional virtualized operating systems. Several cross-layer optimizations are targeted, ranging from hypervisor on host OS up to the guest OS. In addition to improving efficiency, another goal of the project is to simplify deployment of applications in the cloud. MIKELANGELO initially focused only on application package management with its extensions done primarily to the Capstan open source project. A completely new way of composing self-sufficient virtual machine images based on OSv [5] is introduced. It allows for flexible and efficient reuse of pre-built application packages that are readily provided by the consortium. It also builds on best practices on how to approach the package preparation with various packages from the HPC and Big data fields. Since then, progressed towards cloud management and application orchestration. This comprises the integration of full support for deployment of unikernels onto OpenStack. Application orchestration using container-like interfaces is the last step towards management of lightweight applications on top of heterogeneous infrastructures. All of this has been joined under one main toolbox (LEET - Lightweight Execution Environment Toolbox). On top of these components, the project also delivers components spanning over all the layers of the software stack. This includes snap and SCAM. Snap is a holistic telemetry solution aimed at gathering data from all the layers of the target infrastructures, ranging from the hardware through all layers of the infrastructure software to the applications themselves. Snap furthermore provides flexible telemetry data processing capabilities and a wide range of storage backends. Finally, MIKELANGELO uses two commonly used deployment targets: Cloud and HPC. The purpose of integrating the aforementioned components into a common environment is to demonstrate the potential of the individual components. Cloud integration presents the architectural design and gives details on how the integration with OpenStack is achieved. There is the updated CI integration component Scotty and also a new component dedicated to live re-scheduling of resources in Clouds, called Mikelangelo Cloud Manager (MCM). Another tool introduced is called OSmod and is utilized to modify the host operating system. Integration of all components eligible for HPC environments is achieved by extending the widely-spread batch system management software Torque, under the name of vTorque. It extends PBS/Torque by virtualization capabilities to handle VM instantiation, job deployment and VM tear down within the Torque job life cycle in a transparent way to the user.

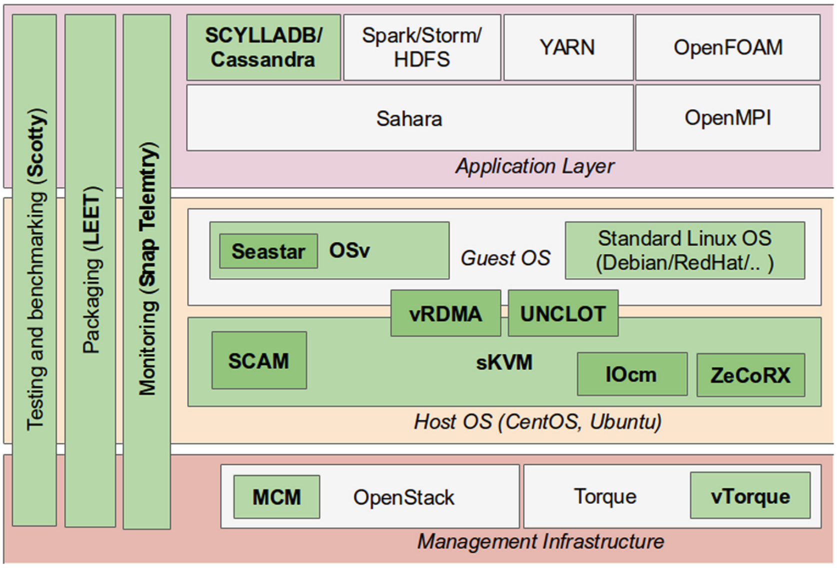

Fig. 1. High-level Cloud and HPC-Cloud architecture.

3 Evaluation and Validation

There are four use cases to validate the project results which provide a perspective to transition project results into active exploitation. The use cases are: big data, aerodynamic maps, cancellous bones and a cloud bursting use case. These cover the most important areas of applications for Cloud and HPC. Cloud computing is tackled by the big data use case, by the aerodynamics use case, and by the cloud bursting use case. Big data is specifically handled by the big data use case. It deployes a virtualized big data platform in the cloud with an automated deployment of synthetic workloads and defined real-world workloads for validation. The cloud bursting use case has developed a new IO scheduler for ScyllaDB, an IO tuning component for ScyllaDB, it improved an RPC framework, and it enhanced the efficiency of data streaming for database replication. Typical HPC applications are covered by the aerodynamic maps use case and the cancellous bones use case. The aerodynamics use case has created a simulation platform with a dashboard supporting the submission of experiments, defined test cases, implemented application packaging. The cancellous bones use case ported its simulation to OSv. The figures below provide an evaluation of accomplished improvements, beneficial to all four use-case, tackling the most crucial part of today's virtualization, the I/O (Figs. 2 and 3).

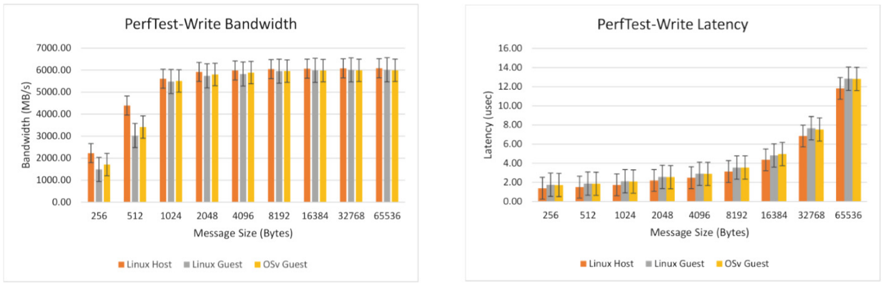

Fig. 2. vRDMA result

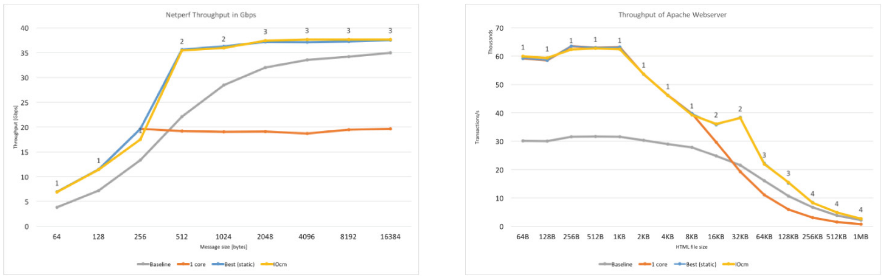

Fig. 3. sKVM I/O

Acknowledgments. The work described in this document has been conducted within the Research & Innovation action MIKELANGELO (project no. 645402), started in Jan 2015, and co-funded by the European Commission under the Information and Communication Technologies (ICT) theme of the H2020 framework programme (H2020-ICT-07-2014: Advanced Cloud Infrastructures and Services).

References

1. Drăgan, I., Fortiş, T.-F., Iuhasz, G., Neagul, M., Petcu, D.: Applying self-* principles in heterogeneous cloud environments. In: Antonopoulos, N., Gillam, L. (eds.) Cloud Computing. CCN, pp. 255–274. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54645-2_10
2. Felter, W., Ferreira, A., et al.: An updated performance comparison of virtual machines and Linux containers. In: 2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 171–172, March 2015
3. HLRS: Inside (2015). http://inside.hlrs.de/editions/15spring.html#mike
4. Adaptive Computing Enterprises, Inc.: Torque resource manager (2017). http://www.adaptivecomputing.com/products/open-source/torque/
5. Cloudius Systems: OSv (2017). http://osv.io/