Linux containers simplify engineering and scientific simulations in the cloud

Wolfgang Gentzsch
The UberCloud Inc.
Los Altos, CA, USA
Wolfgang.gentzschTheUberCloud.com

Abstract— **While many cloud providers today offer powerful computing infrastructure as a service, and enterprises are already making routine use of it, the adoption of cloud computing for engineering and scientific applications is lagging behind. Despite the many benefits cloud resources provide, reasons for this slow adoption are many: complex access to clouds, inflexible software licensing, time-consuming big data transfer, no control over their assets, service provider lock-in, to name a few. But recently, with the advent of the UberCloud's high-performance container technology many of these roadblocks are being removed. In our paper we describe the current status and landscape of clouds for engineers and scientists, the benefits and challenges, and how UberCloud is providing an online solution platform and container technology which reduce or even remove completely many of the current roadblock, and thus offer every engineer and scientist additional compute power on demand, in an easily accessible way.**

Keywords—cloud computing; high performance computing; online marketplace; computer simulations; CAE; computational biology; UberCloud Experiment

I. Introduction

Engineers and scientists today have two powerful options for applying additional computing and computer simulations to their product design, development, and research, in addition to their desktop workstations: high performance computing (HPC) servers, and HPC Clouds. The benefits of using these tools are huge, such as enormous cost savings; reducing product failure early in the design, development, and production; developing optimized processes; achieving higher quality products to keep existing and gaining new customers; and shortening time to market. Potentially, all this can lead to increased competitiveness and innovation.

However, less than 5% of manufacturers are using HPC servers or HPC clouds for computer simulations, according to several studies from the US Council of Competitiveness, [01]. The vast majority (in fact about 95%) perform virtual prototyping or large-scale data modelling still just on their desktop computers. But, 57% of these companies said that they have application problems that they can't solve because their desktops are too slow for the problems they want to solve or because geometry or physics are too complex and need more memory than is available from their desktop.

The first option of acquiring additional compute power is buying an HPC server which is many times faster than what engineers currently have available on their desk. However, for many organizations, especially small and medium size enterprises (SMEs) and small academic departments, buying a large HPC server is often not a viable alternative. In addition to the high cost of expertise, equipment, maintenance, software, and training, there are often long and painful internal procurement and approval processes, and additional skills and manpower are needed to operate and maintain such a system.

The second option for SMEs to experience the benefits from HPC is recently offered by cloud computing. HPC in the Cloud allows engineers and scientists to continue using their own desktop system for daily design and development work, and to submit the larger, more complex, more time-consuming jobs to the cloud. Additional benefits are on-demand access to 'infinite' resources, pay per use, reduced capital expenditure, greater business agility, higher-quality results, lower risk, lower product failure rate, and dynamically scaling resources up and down as needed.

In our full paper, we will present an overview of the status and trend of HPC in the Cloud for the manufacturing and scientific markets. We provide the status and trend of HPC in the Cloud for the Independent Software Vendor market, and look at the competitive landscape of current Cloud resource providers. We include an outlook on how the UberCloud HPC Experiment can accelerate the process of go-to-market and customer acceptance. Finally, we highlight the underlying Linux container technology which reduces or even completely removes most of the current cloud computing roadblocks.

II. Cloud, HPC Cloud, and Cloud Applications

Definition: According to NIST, the National Institute of Standards and Technology [02], Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

The architecture of an HPC Cloud is designed to run specific HPC workloads from various science and engineering application areas. Still standard cloud services can address a portion of the HPC market. Those applications are usually

978-1-4799-8311-7/15 $31.00 © 2015 IEEE
DOI 10.1109/GOCICT.2014.11

embarrassingly parallel (like parameter studies with varying parameter input) and have low I/O requirements (light I/O). Careful analysis of application requirements is needed in order to determine the effective HPC performance in standard cloud offerings. However, many applications developed and optimized for HPC systems require additional system features in the Cloud as well, such as

- large capacity and capability, application software choice, and physical (so-called bare-metal) or virtualized environment depending on performance needs. The use of high performance interconnects and dynamic provisioning can offer cloud features while maintaining HPC performance levels.

- high performance I/O is often necessary to ensure that many I/O-heavy HPC applications will run to their fullest potential. As an example, pNFS might provide a good plug-and-play interface for many of these applications. Back-end storage design, however, will be important in achieving acceptable performance.

- fast network connection between the high performance cloud resources and the end-user's desktop system. Scientific and engineering simulation results are often in the range of many Gigabytes to a few Terabytes. Additional solutions here are remote visualization, data compression, or overnight sending a disk with the resulting data back to the end-user.

Based on these observations, performing HPC in the cloud is indeed possible. Some (mostly 'embarrassingly' parallel) applications perform well on existing standard cloud, but many HPC applications cannot be shoehorned into existing standard cloud solutions. Clouds that are specifically designed for HPC are needed, and indeed represent a viable solution for many organizations, especially if the number of cores required is moderate (often in the order of 16 – 64 CPU cores). In addition, other issues may need to be discussed before an HPC cloud can deliver low-cost and flexible HPC cycles.

III. UBERCLOUD ACCELERATING HPC IN THE CLOUD

Cloud Computing and its emerging technologies, such as virtualization, Linux containers sitting on bare metal, web access platforms and their integrated toolboxes, solution stacks accessible on demand, automatic cloud bursting capabilities, and more, enable research and industry to use additional computing resources in an elastic and affordable way, on demand. Now, the UberCloud HPC Experiment provides a platform for researchers and engineers to explore, learn, and understand the end-to-end process of accessing and using HPC Clouds, to identify the concerns, and resolve the roadblocks, details see [03]. End-user, software provider, resource provider, and an HPC expert are collaborating in a team and are guided through a 22-step end-to-end process, jointly solving the end-user's application problem in the cloud.

The current status, as of September 2014, is quite impressive. 155 teams have been formed around industry end-user applications running on remote HPC center and HPC Cloud resources. Over 2000 organizations from 72 countries are participating. The end-to-end process of taking applications to the cloud, performing the computations, and bringing the resulting data back to the end-user has been partitioned into 22 single steps which the teams closely follow on the Basecamp collaboration environment. An UberCloud University has been founded providing regular educational lectures for the community. And the one-stop UberCloud Exhibit offers an HPC Services catalogue where the community members can exhibit their cloud related services or select the services which they want to use for their team experiment or for their daily work. Many UberCloud Experiment teams publish their results widely, in the meantime, see e.g. article from Sam Zakrzewski and Wim Slagter from ANSYS about: On Cloud Nine [04]. Finally, at the Supercomputing Conference in Denver in November 2013, The UberCloud received the HPCwire Readers Choice Award for the best HPC Cloud implementation, [05].

Let's define what roles each stakeholder has to play to make service-based HPC come together. In this case, stakeholders consist of industrial end-users, resource providers, software providers, and high performance computing experts:

The industry end-user: A typical example is a small or medium size manufacturer in the process of designing, prototyping, and developing its next-generation product. These users are prime candidates for HPC-as-a-Service when in-house computation on workstations has become too lengthy a process, but acquiring additional computing power in the form of an HPC server is too cumbersome or is not in line with IT budgets. HPC is not likely to be the core expertise of this group either.

The application software provider: This includes software owners of all stripes, including ISVs, public domain software organizations and individual developers. The experiment usually prefers rock-solid software, which has the potential to be used on a wider scale. For the purpose of this experiment, on-demand license usage will be tracked in order to determine the feasibility of using the service model as a revenue stream.

The HPC resource providers: This pertains to anyone who owns HPC resources, such as computers and storage, and is networked to the outside world. A classic HPC center would fall into this category, as well as a standard datacenter used to handle batch jobs, or a cluster-owning commercial entity that is willing to offer up cycles to run non-competitive workloads during periods of low CPU-utilization.

The HPC experts: This group includes individuals and companies with HPC expertise, especially in areas like cluster management. It also encompasses PhD-level domain specialists with in-depth application knowledge. In the experiment, experts will work as team leaders, with end-users, computer centers, and software providers, to help glue the pieces together.

For example, suppose the user is in need of additional compute resources to increase the quality of a product design or to speed up a product design cycle, say for simulating more sophisticated geometries or physics, or for running many more simulations for a higher quality result. That suggests a specific software stack, domain expertise, and even hardware configuration. The general idea is to look at the end-user's tasks and software, and select the appropriate resources and expertise that match certain requirements.

Then, with modest guidance from the UberCloud Experiment, the user, resource provider, and HPC expert will implement and run the task and deliver the results back to the end-user. The hardware and software providers will measure resource usage; the HPC expert will summarize the steps of analysis and implementation; the end-user will evaluate the quality of the process and of the results and the degree of user-friendliness this process provides. The experiment orchestrators will analyze the feedback received. Finally, the team will get together, extract lessons learned, and present further recommendations as input to their case study.

IV. WIND TUNNEL FLOW AROUND BICYCLE AND RIDER

As a glimpse into the wealth of UberClouds 155 experiments so far, here is a short summary of Team 58: Wind Tunnel Flow around Bicycle and Rider.

The team consisted of end-user Mio Suzuki from Trek Bicycle, software provider and HPC expert Mihai Pruna from CADNexus, and resource provider Kevin Van Workum from Sabalcore Cloud Computing.

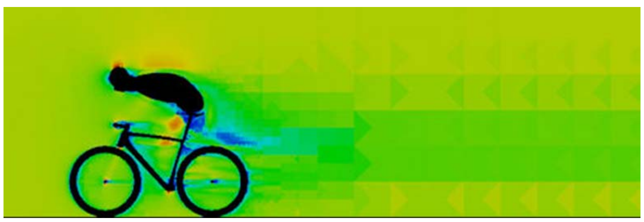

Fig. 1. Team 58 - Velocity color plot, generated with CADNexus Visualizer Lightweight Postprocessor.

The CAPRI to OpenFOAM Connector and the Sabalcore HPC Computing Cloud infrastructure were used to analyze the airflow around bicycle design iterations from Trek Bicycle. The goal was to establish a great synergy among iterative CAD design, CFD analysis and HPC Cloud. Automating iterative design changes in CAD models coupled with CFD significantly enhanced the productivity of engineers and enabled them to make better decisions. Using a cloud-based solution to meet the HPC requirements of computationally intensive applications decreased turnaround time in iterative design scenarios, and reduced the overall cost of the design. The complete case study from Team 58 can be found in the 2013 Compendium of case studies [03].

V. LINUX CONTAINERS

UberCloud Containers are ready-to-execute packages of software. These packages are designed to deliver the tools that an engineer needs to complete his task in hand. The ISV or Open Source tools are pre-installed, configured, and tested. They are ready to execute, literally in an instant with no need to install software, deal with complex OS commands, or configure.

The UberCloud Container technology allows wide variety and selection for the engineers because they are portable from server to server, Cloud to Cloud. The Cloud operators or IT departments no longer need to limit the variety, since they no longer have to install, tune and maintain the underlying software. They can rely on the UberCloud Containers to cut through this complexity.

This technology also provides hardware abstraction, where the container is not tightly coupled with the server (the container and the software inside isn't installed on the server in the traditional sense). Abstraction between the hardware and software stacks provides the ease of use and agility that bare metal environments lack.

A. The Underlying Technology: UberCloud Containers

The UberCloud Container technology allows wide variety and selection for the engineers because they are portable from server to server, Cloud to Cloud. The Cloud operators or IT departments no longer need to limit the variety, since they no longer have to install, tune and maintain the underlying software. They can rely on the UberCloud Containers to cut through this complexity.

UberCloud Linux Containers are the underlying technology for the UberCloud Marketplace, [06]. They rely on Linux kernel facilities such as cGroups, libcontainer, and LXC which are already a part of many modern Linux operating systems. The run time components to launch UberCloud Containers are distributed as a popular open source product called Docker and don't require an additional capital investment. We enhanced Docker [07], [08], to be well-suited for technical computing applications in science and engineering. Docker can package an application and its dependencies in a virtual container that runs on any Linux server. This helps enable flexibility and portability on where the application can run.

Docker automates the deployment of applications inside software containers by providing an additional layer of abstraction and automation of light-weight operating system–level virtualization on Linux. Docker uses resource isolation features of the Linux kernel such as cgroups and kernel namespaces to allow independent "containers" to run within a single Linux instance, avoiding all the heavy overhead of starting virtual machines.

Linux kernel's namespaces completely isolate an application's view of the operating environment, including process trees, network, user IDs and mounted file systems, while cgroups provide resource isolation, including the CPU, memory, block I/O and network. Docker includes the libcontainer library as a reference implementation for containers, and builds on top of libvirt, LXC (Linux containers) and systemd-nspawn, which provide interfaces to the facilities provided by the Linux kernel.

UberCloud Containers are launched from pre-built images which are distributed through a central registry hosted by UberCloud. Software and operating system updates, enhancements, and fixes become instantly available for the next container launch in an automated fashion.

The notion of a pre-built image may sound familiar; this notion has been at the heart of virtualization, a popular technology for logically breaking down a physical computer environment into finer pieces. However, unlike virtualization, UberCloud light-weight Containers don't rely on a hypervisor; instead, they share the host operating systems kernel and application libraries, leading to performance characteristics that are comparable to bare metal installations.

B. UberCloud Containers: What's inside?

The Short answer is: everything that an engineer needs to complete a particular task on a remote server as if it was running on his desktop. Let's get into further details. UberCloud Containers come complete with:

- **An up-to-date Linux OS.** UberCloud supports a number of popular operating systems and makes its containers available in flavors such as Ubuntu and CentOS. The end user may select one or the other based on her own preference. Linux OS patches are applied periodically by UberCloud, distributed automatically and used when a new container is launched.

- **Libraries commonly needed by engineers**. A good example is MPI, which comes installed and configured inside the container when it's launched.

- **Utilities needed by engineers**. UberCloud provides a well thought out user experience and provides utilities such as screen sharing applications, diff utilities, and remote visualization applications.

- **Cloud data storage connectors**. Many IT departments are opening their doors to cloud data storage and collaboration tools, such as Box.com, and Amazon S3. UberCloud offers connectors to multiple providers, making data transfers from these services a breeze.

- **Engineering applications**. Engineers rely on sophisticated and highly specialized application software to perform their computations. An increasing number of these applications come pre-installed, tuned and tested within UberCloud Containers. A great example is the UberCloud OpenFOAM Container, which comes as a ready to run installation of the latest version of the fluid dynamics OpenFOAM software. UberCloud is in collaboration with distributors of open source as well as commercial applications to further extend its list of supported applications.

- **Administration layer**. Administrative tools such as automated password generation, performance checking, and log monitoring come bundled with the UberCloud Containers. These tools provide the IT staff with an automated process to run and manage the containers.

- **Many hours of testing.** Through hours of testing performed by multiple engineers, the UberCloud Containers evolve to contain less user experience problems and nuisances. The tests are performed using realistic models by a multidisciplinary testing team. The test results are fed right back to the development process, leading to a well-tuned, well tested user experience.

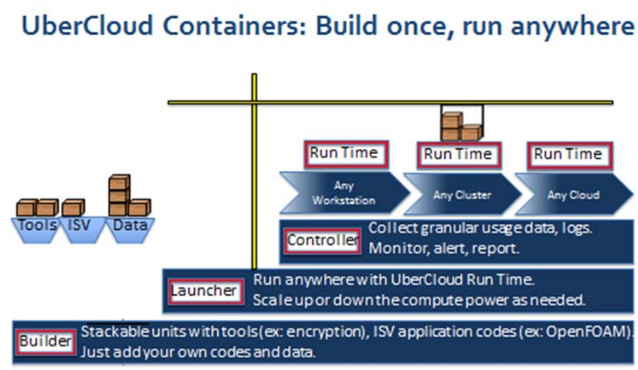

Fig. 2. UberCloud containers: build once, run anywhere.

In Summary, the advantages of using UberCloud Docker containers for accessing and using remote computing resources in the cloud are: software portability, manageability, variety, low overhead, instant provisioning, high resource utilization, and the ability to perform IT audits of the components, configurations, and security settings of the UberCloud Containers. The cloud computing roadblocks which are reduced or even removed by our UberCloud containers are summarized in Fig. 3.

Challenge *)	Addressed today	With UberCloud
Portability	low	high
Security	medium	high
Software Licenses	low	medium
Data Transfer	low	medium
Compliance	low	medium
Standardization	low	high
Cost & ROI Transparency	low	high
Resource Availability	medium	high
Transparency of Market	low	high
Cloud Computing Expertise	low	medium

Fig. 3. List of cloud comuting challenges which are addressed by UberCloud containers. Cloud challenges are addressed low, or medium, or high

References

[1] Council of Competitiveness, 'Make', 'Reflect', 'Reveal', 'Compete' studies 2010. http://www.compete.org/

[2] NIST Cloud Definition: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3] The UberCloud HPC Experiment: Compendium of Case Studies sponsored by Intel, 2013 and 2014: https://www.theubercloud.com/ubercloud-compendium-2013/

https://www.theubercloud.com/ubercloud-compendium-2014/

[4] Sam Zakrzewski, Wim Slagter. On Cloud Nine. Digital Manufacturing Report. April 22, 2013. http://www.digitalmanufacturingreport.com/dmr/2013-04-22/on_cloud_nine.html

[5] The UberCloud Receives Top Honors in 2013 HPCwire Readers' Choice Award, November 27, 2013. http://www.hpcwire.com/off-the-wire/ubercloud-receives-top-honors-2013-hpcwire-readers-choice-awards/

[6] The UberCloud Marketplace with samples of containerized application solutions: https://www.TheUberCloud.com/Marketplace

[7] Chris Swan: Docker: Present and Future. http://www.infoq.com/articles/docker-future

[8] Docker: http://en.wikipedia.org/wiki/Docker_(software)