

Licensing in the cloud

Mehdi Sheikhalishahi, Lucio Grandinetti, and Wolfgang Gentzsch

In this report, we go over some software licensing issues in the context of UberCloud HPCEXperiment (<http://www.hpcexperiment.com>) and cloud computing. Then, we enumerate some solutions for these issues. The solutions exploit software alternatives, or cloud computing landscape to offer the best matching solution. In particular, we propose the following approaches:

- Free Software
- Outsourcing characteristics of cloud computing
- Private or hybrid cloud models
- Cloud based licensing model
- Negotiation and Agreement
- Cloud computing services
- License as a Service
- Provisioning ability of cloud
- Integration of software stacks

Free software

The first solution to licensing problem is to shove off commercial software, and move exclusively to alternative free software solutions.

Cloud computing is definitely a boost for open source which has flexible licensing. There is no question that cloud computing only strengthens the open source value proposition. But open source is not right for every organization, and there will continue to be opportunities for commercial application software even as we transition to the cloud.

From HPCEXperiment Round1-Final Report, we have the following attempts to address licensing issues in the cloud by transitioning toward free software.

“1) To promote the use of free software. 99% of our applications are *free software*. We are now running a project to provide an *open software based CAD/CAE* cloud solution for SMEs in the euro region.”

“**Using the equivalent software and operating system** to resolve licensing issues. Yes, licensing has been a big problem for us too. Regarding Red Hat and Grid Engine, we don't have licensing concerns. Our Red Hat edition is Enterprise Linux HPC and our legal department informed us that our EULA doesn't limit its use only to academic uses.”

Operating System

“Most of the OS requirements that I see are based on Linux. Here we have distributions such as RHEL, Suse and others that you can purchase. At the time I started, our RHEL distribution was so outdated, I actually found it even useless to use such an OS, although it would probably be fine for the majority of application users (we have since than an up to date version). Based on that issue I started early to look into alternatives and I used with a team here *free alternatives to RHEL* which worked better due. Furthermore, I found at that time that using a licensed version of RHEL does not work well as they did not support the more interesting cloud environments we explored. We disbanded for that reason any development in licensed OS versions”

Queuing System

“I had the same experience with the queuing system that is bound towards technologies such as the *free XCAT* and found that, although providing perfectly good solutions for RHEL, they did not support the OS easily we are interested in. We spent some time and effort to hack XCAT to make it work for us. We came at the end to the conclusion that the queuing system and its dependency on HPC software stands in the way of making quick progress on other developments we are more interested in.”

Outsourcing characteristics of cloud computing

When we cannot resolve licensing issues, we may go to the alternative cloud solutions in the cloud market. We may outsource some parts of our software architecture and system to those cloud providers that address our licensing issues. In cloud, instead of thinking about database, application server, or Web server clusters that support scalable Web applications, we can now think about elastic tiers that change dynamically as needed.

Obtaining application functionality via SaaS models also addresses the licensing issue, but SaaS is not for everyone. Some organizations are worried about the lack of predictability inherent in pay-as-you-go pricing, while others require customized or bespoke solutions that are not economical for vendors to offer via a SaaS model. Even with open source and SaaS, there will continue to be a market for commercial enterprise application software that you actually need to install, though into an elastic IaaS cloud environment.

“We have been implementing distributed computing for twenty years now, after all, no longer are applications ”software running on a box,” but rather ”software running on many boxes in multiple tiers.””

Private or hybrid cloud models

For the following scenario in UberCloud HPCExperiment, a private or a hybrid cloud solution would address the problem. The solution is to host part of customer software *that cannot be copy out of the company building* in the private cloud.

“Others types of commercial software are the true problem. We have had several failed projects with companies because software ISVs didn't allow to install *customer software copy out of the company buildings*. We are part of the public sector of a regional government here. With the current economic crisis in our country, we can't afford to buy commercial software and those projects have failed.”

“Hybrid Windows/Linux Cloud Architecture #3 is slightly modified. The *License Server remains in the USA* however *all solver and simulation systems are relocated to Amazon EU region in Ireland* for performance reasons. End-user switches all simulation work to inexpensively sourced nodes from the Amazon Spot Market. The “Modified Design #3” in which solver/simulation systems are running on AWS Spot Market Instances *in Ireland while a small license server remains in the USA* reflect the final “design”.”

Cloud based licensing model

On-demand, short-term, and pay-as-you-go licensing models are cloud computing licensing models. We would suggest customers to not to enter into long term contracts in terms of volumes and time period.

From HPCExperiment Round1-Final Report, we have the following recommendations and attempts to address licensing issues in the cloud by Cloud based licensing model.

“Many software providers are already working on compatible licensing models. We recommend end-users to contact their software providers early on and include them in their HPC initiatives. Alternatively, the HPC Experiment is a great way to work hand-in-hand with a resource provider on testing an HPC in the Cloud

compatible licensing model. There are already existing successful *on-demand licensing models* from some forward looking ISVs (also part of our experiment) which we believe the others can learn from.”

“Having short-term licenses which scale with the need of a simulation greatly reduces our costs by preventing the purchase of under-utilized HPC packs for our company’s in-house simulation.”

“We looked closely at how much friction was caused by software licensing models of the providers which were willing to participate in the Experiment. Although many software providers are diligently working on making on-demand licenses available, the landscape is difficult to navigate. It’s hard to predict which software provider has on-demand licensing models fully developed and it’s even more difficult to know how to work with their significantly different models. Many of our teams ran into license management challenges and required extra help to move forward.”

Negotiation and Agreement

All involved parties need to negotiate to make agreements with each other to develop a suitable licensing model for cloud computing based on many parameters. These involved parties include ISVs, software and hardware companies, industry partners, and end-user.

“Yes, this is a challenge. We run CentOS on our clusters so there is no issue with users getting on the 'bare' system; it is *the ISV software that becomes an issue*. In *our agreements with external users*, we state that the user must acquire the appropriate commercial licenses, or we will acquire them on the user's behalf and invoice the user for the cost. What this does is introduce some 'friction' in project startup because licensing costs must be negotiated, purchased, etc. If *ISVs can move to on-demand/pay-as-you-go types of licensing* arrangements, the situation will improve. I have had some success with XYZ, who *permitted us to purchase a monthly license* for their software to support a *commercial customer*. We renewed the license monthly and included the cost in our monthly invoice to the customer, which worked out pretty well for the project. This is a major issue that has no easy answers at the moment, but I believe it will improve over time”

“We are now working with models in which we can have preinstalled the applications, and provided that the industry user has an *agreement* with the ISV, and paying if needed of course. What we found is that the ***ISVs want to have direct contact with the industry user***, and recently, we see that they are more open to collaborate with us.”

“It is clear that without the participation of the software vendors in developing compatible software licensing models, the adoption of HPC in the Cloud will be significantly slower.”

“As a case in point, we’ve noted one project where the software provider wasn’t willing to participate in the Experiment with their end-user customer. Since the requirements of the end-user were tightly coupled with the software they were using the end-user needed to terminate the project.”

Cloud computing services

“Our design pattern for the commercial license server is potentially a useful best-practice. By leveraging custom/persistent MAC addresses via the use of Elastic Network Interfaces (ENI) within Amazon VPC we were able to build a license server that would not “break” should the underlying hardware characteristics change (common on the cloud).”

“Licensing with dynamic resources requires putting license servers in the data-center.”

License as a Service

In the cloud, license management may be provided as a service. For that, a user request a license by sending its complete information and software stack to License Management Service. It is the responsibility of this service to negotiate with all parties, i.e., ISVs, software and hardware companies, industry partners, cloud infrastructure providers, cloud service providers, and end-user, and based on user's information provide a suitable license.

“We looked closely at how much friction was caused by software licensing models of the providers which were willing to participate in the Experiment. Although many software providers are diligently working on making on-demand licenses available, the landscape is difficult to navigate. It’s hard to predict which software provider has on-demand licensing models fully developed and it’s even more difficult to know how to work with their significantly different models. Many of our teams ran into license management challenges and required extra help to move forward.”

Provisioning ability of cloud

“We also offer a queuing system for our HPC systems. I will review the licensing of that one, but as far as I know this can be used for research even if it is conducted by industry. However, also here we could completely replace the queuing system with for example SGE if this is desirable. Recently we even got a request from one of our user to do so. Certainly that is possible, if we would have enough manpower. However, we gave the recommendation why not provision one of the worker nodes as the manager for SGE and then provision some workers that you register to this management node and you are all set. In the matter of fact we did work towards this goal and I can report to you an intermediate result. We have the ability to set up a virtualized queue management system (which currently does not run as we upgrade to OpenStack Folsom) but allowed a user (using OpenStack Essex) to set up his own queue with a single command.”

“e) As we also support bare metal provisioning the same image template that can be staged into an IaaS framework could at one point also be staged directly onto the server nodes, thus ending up in a "private" bare metal cluster for the users. Naturally as you can see, this could mean it could *carry its own licenses* if that is supported by the underlying software. However I realize this is on paper easier than actually done in reality as many license models require a registration per hardware and not per use.”

“f) As you can see our environment is not just an IaaS but it provides a very different aspect: *Assemble your own Infrastructure* that suites your needs best. It goes far beyond the question what is the best IaaS framework for a user or how reliable and easy is the access to such IaaS for a scientist. As you can see what we do are very advanced definition of software systems that are used by advanced engineers.”

Integration of software stacks

There are many software components, tools, and operating systems to build a software system. Their licenses agreements must integrate with each other. In the licensing landscape, we are in the need of license validation tools to check and verify this integration.

Operating System, HPC tools: Resource Manager, Scheduler, HPC application, and software tools

Output: tools to validate the integration and the combination of OS, RM, and scheduler, software tools