NEON – Northern Europe Cloud Computing

Final Report

December 17, 2010

Åke Edlund and Maarten Koopmans

Zeeshan Ali Shah, Ilja Livenson, Frederik Orellana, Jukka Kommeri, Miika Tuisku, Pekka Lehtovuori, Klaus Marius Hansen, Helmut Neukirchen, Ebba Þóra Hvannberg

NEON – big picture of piloting environment

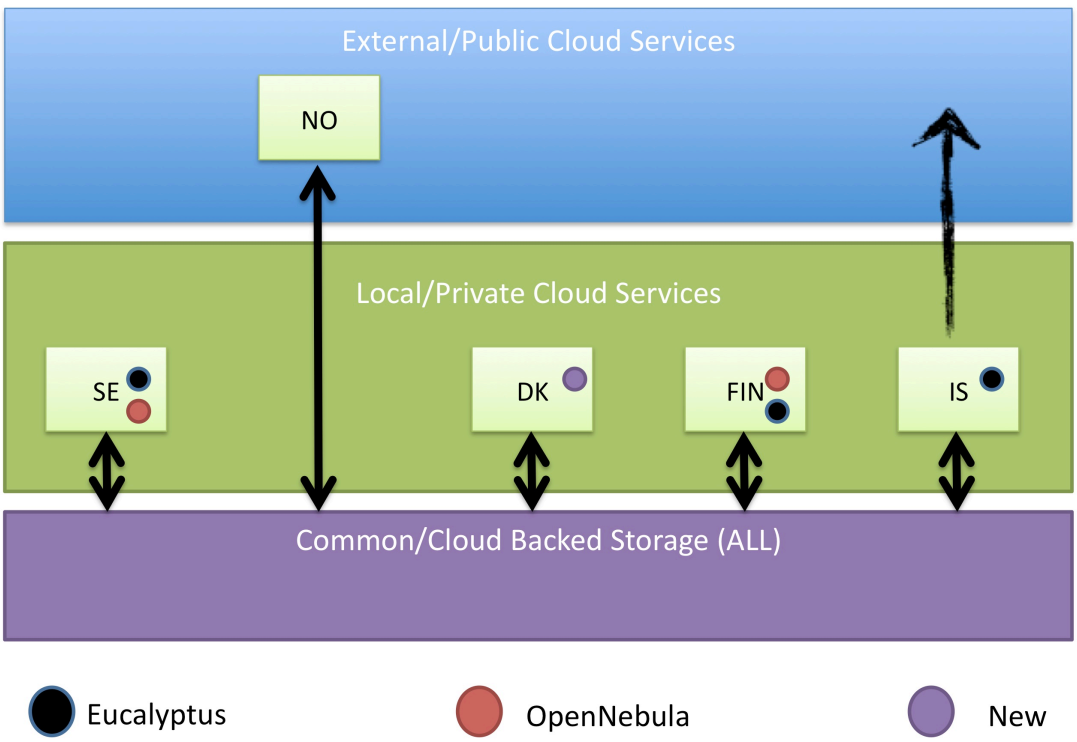


Table of Contents

1 About the Report

Main Scope – Answer the questions "Why", "If", and "How" use cloud computing?

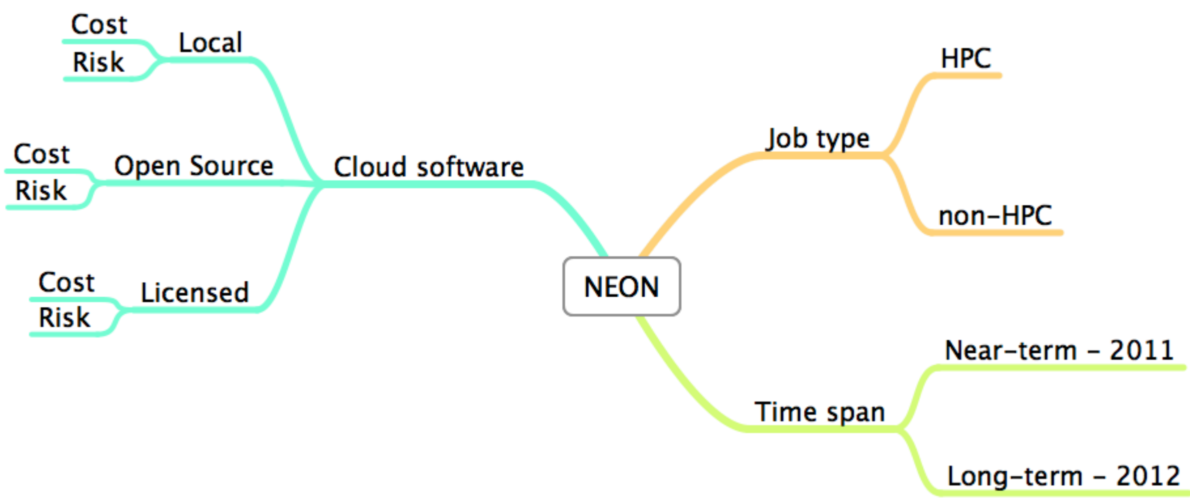
The NEON project goal was to produce a report describing: state-of-the art of cloud computing; cost of moving and running non-HPC jobs on a cloud computing environment; how to do this in practice; a list of identified risks/benefits on a short/long perspective.

Additional Efforts

In addition the project did an educational and knowledge spreading effort in the Nordic region, reported in the Appendix of this report. A knowledge portal was established during the project, with input from many similar European cloud projects: www.scientific-cloud.org

Higher Level

The higher ambition with the project report was to gather decision information to policy bodies such as eIRG to help with the strategic long-term cloud plans for Europe.



The NEON project focused on how/if to move non-HPC jobs to cloud resources

2 Executive Summary and Recommendations

The findings of this project are:

1. *Private cloud technology is not mature enough yet to provide a transparent user experience. It is expected that this will be the case mid 2012. The cost effectivity of both public and private cloud should be continuously monitored as there is a strong downward trend. This conclusion is supported by NEON experimenting as well as larger initiatives e.g. StratusLab report (add link).*
2. *Public cloud technology is mature enough but lacks certain features that will be necessary to include cloud resources in a transparent manner in a national infrastructure like Notur (e.g. quota management). These features are emerging in 2011 via third party management software and in th ebest of breed public cloud services.*
3. *Public clouds are competitive in the low end for non-HPC jobs (low memory, low number of cores) on price*
4. *A significant fraction (ca. 20%) of the jobs running on the current supercomputer infrastructure are potentially suitable for cloud-like technology. This holds in particular for single-threaded or single-node jobs with small/medium memory requirements and non-intensive I/O.*
5. *There is a backlog of "real" supercomputer jobs that suffers from the non-HPC jobs on the supercomputer infrastructure. Off-loading these non-HPC jobs to a public cloud would effectively add supercomputing capacity.*
6. *Available storage capacity is not accessible in a user-friendly way; most storage clouds are only accessible via programmable interfaces.*
7. *Continue the participation in the ECEE[2] to contribute to the European roadmap for eScience and Clouds.*

Note: the list above has been distributed to the ECEE[1] for feedback. The feedback received supported these findings. The ECEE is a collaboration effort lead by NEON coordinating knowledge sharing between European cloud projects including NEON, BalticCloud, NGS (UK), GRNET cloud, SARA cloud (NL), UCM (OpenNebula), StratusLab, VENUS-C, SEECCI (Balkan) and CESGA (Spain, part of Open Cirrus). The overall comment was that they all agree to points 2-6. 7 was not in the list when we mailed them), with examples of successful private clouds (i.e. point 1.

[1] www.scientific-cloud.org

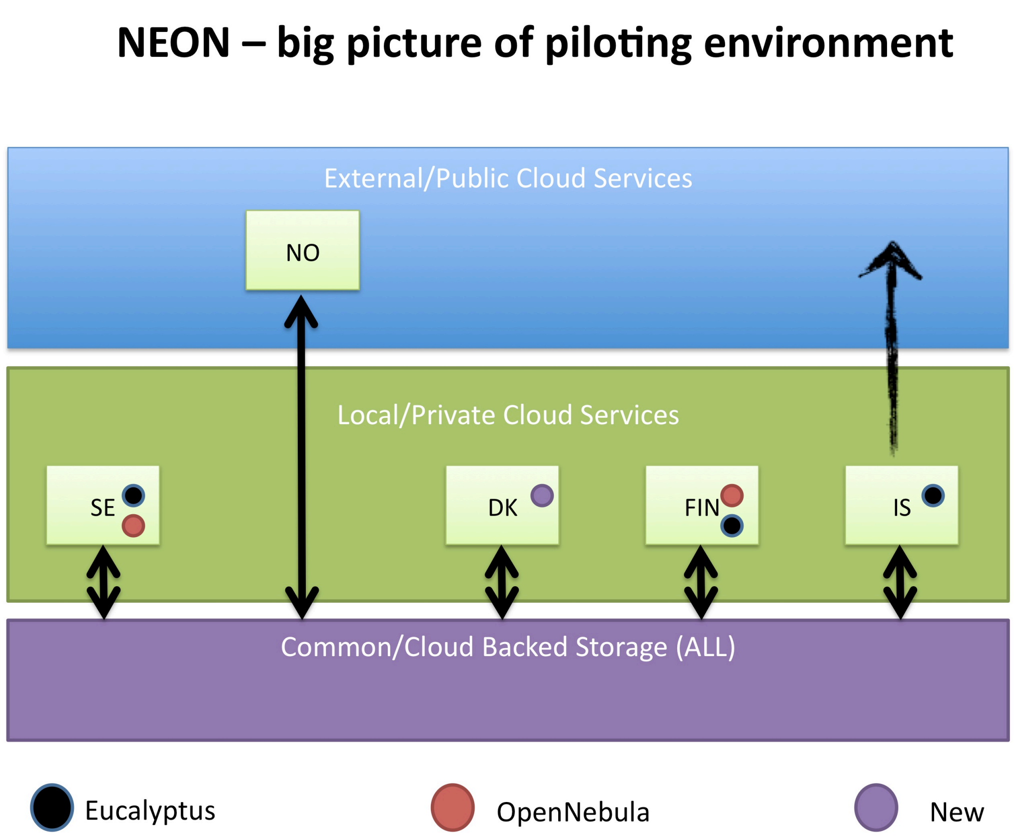
2.1 Short Term Recommendations

Private clouds are not mature enough for our users, public clouds are

Cloud computing today offers already a value, both in cost and in usability, for the non-HPC user. Private cloud services are still at an early stage, and not easy to use for a regular administrator. While Private clouds (e.g. Eucalyptus, OpenNebula) have some way to go, Public clouds, especially from Amazon, are now in a more mature stage: well documented, easy to use, predictable and feature rich. In addition, Amazon has a number of initiatives for academic use, e.g. the Amazon Education program[2] with recurring grants for research applications. In the NEON project we applied for one of these (USD 5,000), and got the application accepted within two weeks - and started to use it the same day.

Many pilot installations, one bioinformatics pilot, and a common storage layer to connect all together

During the project a number of alternatives were tested (see picture below) and in addition a common open source cloud-backed storage service was set up to link these together. A pilot in bioinformatics, eSysbio, was launched on a public cloud with promising results (mostly based on the above mentioned Amazon grant).



[2] http://aws.amazon.com/education

Cost: Public clouds are on par with local alternatives. Private clouds are not predictable

Looking at the cost of using clouds, we focused on the Public side comparing a real-life HPC cluster with a non-HPC cluster offering from Amazon. This comparison gave at hand that the costs are comparable, with a higher flexibility on the Amazon alternative. Private clouds are still too hard to install, manage and maintain – making the cost calculation futile. There are management tools, not for free, like RightScale, that mitigate this. The cost of RightScale and similar tools was considered too high for our community, and it would still not remove the current issues with the private clouds' immaturity level.

Note: we did manage to install private clouds on a number of sites but the experience shows that the work and support needed was too high to be of practical usage for large scale deployments without significant extra manpower.

Risks: Lock-in effects continue to be an issue, limiting the usage for some researchers

Lock-in effects are the top risk when using both the public and private cloud. This has to be evaluated on a case-by-case basis, emphasizing the limitations in publishing private/sensitive data for public clouds. Specifically, data transfer costs for larger deployments may cause an economic lock-in when moving away from a public cloud. Private clouds lock-in is migration cost; given the maturity level it is likely that cross-grades need to take place because of enhanced and new (required) features, adding extra cost for manpower.

Near-term Recommendation – use Public clouds for non-HPC and some HPC users

Due to the immaturity of the private cloud offerings, we recommend users to take into account the need of advanced system administrators to install, use and manage private clouds of today. If not comfortable with these services, better wait for them to mature and meanwhile focus on public cloud offerings, and/or on improving the local virtualization efforts. This way, an organization can become familiar with the underlying technology and build operational excellence with regards to cloud technology.

Public clouds are ready to be used, as e.g. the example of eSysbio shows. There are many ways and services for simplified public cloud deployments, e.g. like Heroku, Rightscale or Amazon's own Elasticfox or web based management console.

Next valuable input from real-life-testing: NOTUR Cloud service 2011

Support of the conclusions above can be found in the Norwegian initiative for 2011 with the following strategy: *"--- Notur infrastructure should actively try to start moving non-HPC jobs to cloud technology in 2011. Initially this should be done by offering a cloud computing service on a small scale in a public cloud. This will lead to organizational and operational experience and excellence and bootstrap and organize the user communities. After 2011 a private cloud may be set up or a public cloud can still be used – depending on maturity and pricing of both types of clouds after 2011."*

> *One recommendation is to follow the Notur 2011 service on how they will use cloud computing for their users.*

2.2 *Long-term Recommendation*

Wait and learn, continue testing public cloud offering, cooperate internationally - take lead on public cloud

'Wait and see' is sometimes a good option, but in this case it is better to 'Wait and learn', i.e. continue with the above public cloud experiments by deploying a small-scale cloud service for non-HPC jobs while this new field matures. This is similar to what is described above for NOTUR 2011. Running non-HPC jobs, and even smaller HPC jobs, on a public cloud, learning more about very long-term stability and cost fluctuations – and mobility – is what we recommend. The eSysbio pilot started in July and has been running since, giving good first input for a bigger 'Virtual Data Center' in the public cloud.

Another development during the 'wait and learn' period could be to establish a cloud backed storage usable for all cloud users, independent of cloud focus. This storage service would be the 'glue' of a cross-nation wide cloud service. Alternatively, current storage resources could be opened up as a private cloud storage solution, as deploying storage only is inherently less complex than deploying a complete cloud stack.

Following the above near-term recommendation is an adaptive way forward, and together with ECEE a way to minimize risk of double work, and repeating of mistakes already done by others. In addition the ECEE roadmaps gives us a say and insight into future common projects and collaborations - as well as possible interoperability challenges.

1. *Following the above near-term recommendation is an adaptive way forward, especially if based on ECEE and other international collaborations. [Mitigates the risk of not being part of the future evolution of clouds for eScience]*
2. *Stay ahead of the expected user adoption to public cloud offerings - by being the primary point of contact for any researcher wanting to use e.g. Amazon. This is achieved by creating a shared knowledge among the NGIs on how to best use Amazon and others, and in addition by being a preferred customer for e.g. Amazon - getting better support and consolidated (cheaper) pricing than the direct use of Amazon. In addition we could be the 'grants office' for e.g. Amazon in our region/countries. [Mitigates the risk of losing users and users losing time and money repeating (by us) known mistakes]*
3. *Keep a small scale private cloud up to date and monitor the feature set, complexity and cost during 2011; determine a go/no-go for private clouds versus public clouds at the end of 2011. [Mitigates the risk of slow start in adopting clouds for eScience]*

2.3 *Risks and their Mitigation*

Following above recommendations mitigates the risk of losing the initiative in cloud usage for eScience in our region. I.e. if we do nothing on clouds, the users will, presumably on clouds. The rationale is quite simple: for small to medium sized projects computing cost is small in absolute numbers. Research that has not been awarded "time" might find other ways to get enough money to start n a cloud. This will result in a less structured and cost efficient usage of cloud for eScience. Also, the initiative might be lost.

Another risk is the described sensitivity of data and usage of clouds off premises, depending on local legislature. When using clouds special care must be taken to see if the organization participates in the safe harbor program if it is US based[3].

Finally, there is a risk of economic lock-in for private and public clouds and economic DoS for public clouds. As described above, the economic lock-in mostly concerns data transportation costs for public clouds and a high-frequent update cycle and its associated costs for private clouds. The economic DoS is unlikely but potentially devastating: if a set of users accounts of a public cloud should be compromised the providing organization (i.e. the NGI) would end up paying a lot of extra money. Note that most public cloud providers do have thresholds built-in that require manual intervention to scale beyond, but the risk is on the providing site.

The following table provides a summary of the risks and the measures:

Risk	Measure
Doing nothing	Users will, and may disrupt current practices and administration
Data sensitivity	Local data storage, only computing in cloud. Safe harbor policy when dealing with US based cloud providers.
Economic lock-in private clouds	Wait and watch, via e.g. ECEE
Economic lock in public clouds (data)	Volume deals and tenders. Keep data local when possible.
Economic DoS	Negotiate volume thresholds; active monitoring tools that monitor overall usage and cost structure.

[3] See http://www.export.gov/safeharbor/

2.4 *Summary Northern Cloud for eScience*

Through the NEON project, the members of NDGF are up to speed on what cloud can deliver for eScience. In addition the Nordic project has been visible also on the international arena, especially through the leadership of the ECEE collaboration. Following the recommendations above, continuing the close international collaboration and continuing the adoption towards clouds for applicable areas we have a careful, but not too careful, way forward into this new way of supporting eScience with compute and storage services.

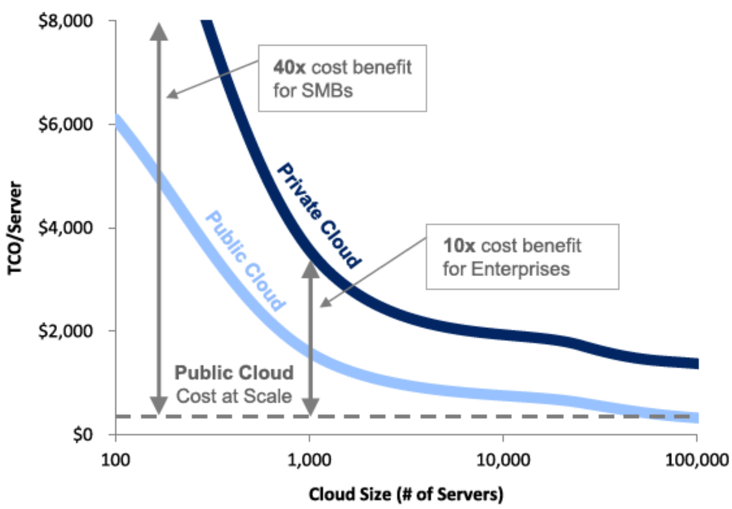
3 The Cost of using Cloud Computing

Economies of Scale and Flexibility in Use

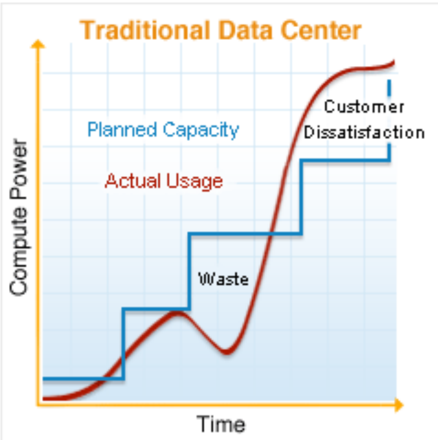
The larger a data center is, the more value technologies such as virtualization and multi tenancy and efficient energy choices can bring. Very large data centers (~50,000 servers), like the one below from Google in Oregon, can be up to 7 times [15] more efficient on administration, and 5 times (US numbers, [15]) more cost efficient on energy compared to mid-size data centers (~1,000 servers). For smaller data centers these differences are even larger. Through this economy of scale big corporations - like Google, Amazon, Microsoft and Facebook - deliver compute and storage services on demand with a competitive price model.

For customers using these cloud services a new flexible way of doing IT evolves - a service based economy - where the user only pays for what they use, and when they need the service. To the right is a recent (November 2010) calculation from Microsoft [19] on the cost benefits when using cloud either using external resources (public clouds) or internal resources (private clouds). As can be seen from this figure there is a clear cost benefit in using cloud resources for smaller up to larger size cases, with a larger value in using public offerings. We'll come back to this picture later on. To decide which service to use the user need to consider a number of things: for how long will the cloud service be used, what size of service, and usage pattern. Usage pattern is hardest to predict, i.e. will the usage be evenly distributed over the period or will there be temporary peaks in usage.

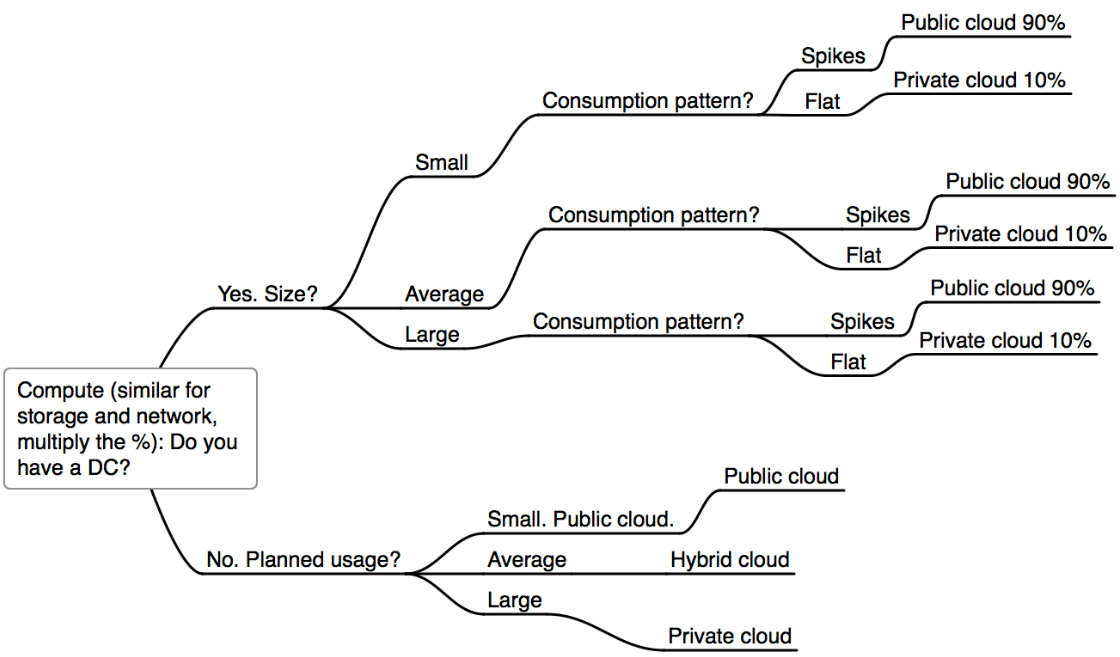
FIG. 22: COST BENEFIT OF PUBLIC CLOUD



Amazon describes [4] the benefit in using clouds in relation to usage pattern as below. In the picture on the left we see a typical behavior of a non-cloud service where the provider, the traditional data center, either over provision or under provision its resources. When over provision the user pays too much for the delivered service. In the case of under provisioning the user of the service is either rejected due to lack of resources or affected by the overload of the resources. In the right side picture a cloud service is described where the over provisioning is small and following the need in a flexible and scalable way.



To help the user of the cloud resources to decide which service to use we[5] have constructed a guiding decision tree model below.

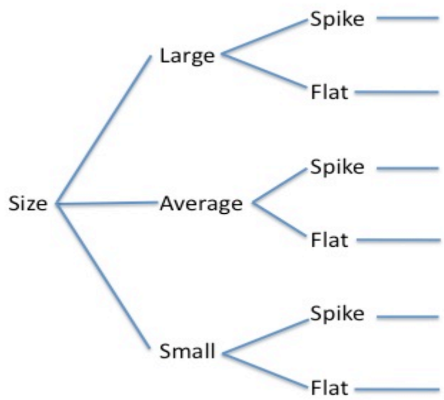


---

[4] http://aws.amazon.com/economics

[5] Ilja Livenson, Åke Edlund - internal report, KTH-SICS Cloud Innovation Center

In this decision tree the user is asked to consider size of use and usage pattern, resulting in below check list.



Compute	Storage	Network
Hybrid	Hybrid	Hybrid
Private	Private	Private
Public	Public	Public
Private	Private	Private
Public	Public	Public
Public	Public	Public

If the user for example wants to conduct work of the size of a large data center with a with a flat predictable behavior on compute, storage and network usage - the optimal choice is to deploy a private cloud on the user's own resources. If on the other hand, for the same case, there is an expected spike in usage of compute resources a hybrid (a private cloud using public cloud resources when needed) is to be considered.

Above reasoning does not consider security need of the user's data. For many applications, e.g. studies of consensus and medical data, public clouds are not allowed due to legal regulations.

Time of usage is another factor to take into consideration. If the usage level is high and over longer time (what is called 'Flat' above) there are break points when a private solution is more cost efficient. Some early studies [1] suggest to use clouds if the total need of compute time is below 12 months, and extra storage need below 6 months. These break points are moving with the decreasing pricing (e.g. 15% price reduction between June 2010 and November 2010) and in the [1] study examples of collaboration between federated private clouds show how these break points can be adjusted.

Cloud computing and environmental costs

Electricity cost is rapidly rising to become the largest element of total cost of ownership, currently representing 15%-20%.

Power Usage Effectiveness (PUE) describes how much extra power is needed to deliver the requested IT service. If PUE equals 1, this means that all inserted power is used to deliver the requested service. PUE equal 2 means that for each kW added for compute, storage and network one more kW is lost on (mainly) cooling. PUE does not show the full picture but gives us an idea of how well the data centers are doing with respect to energy consumption. High PUE renders high financial and environmental cost.

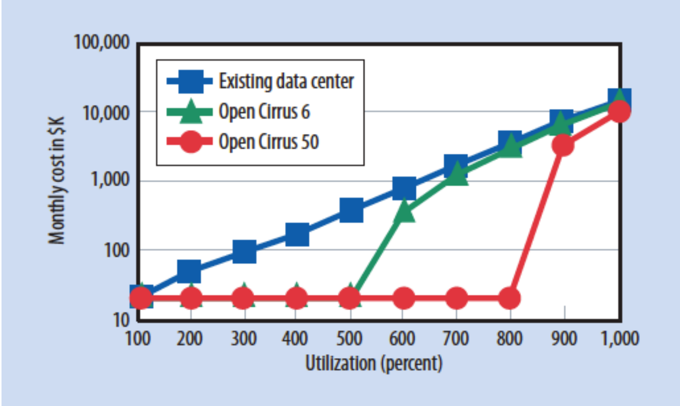
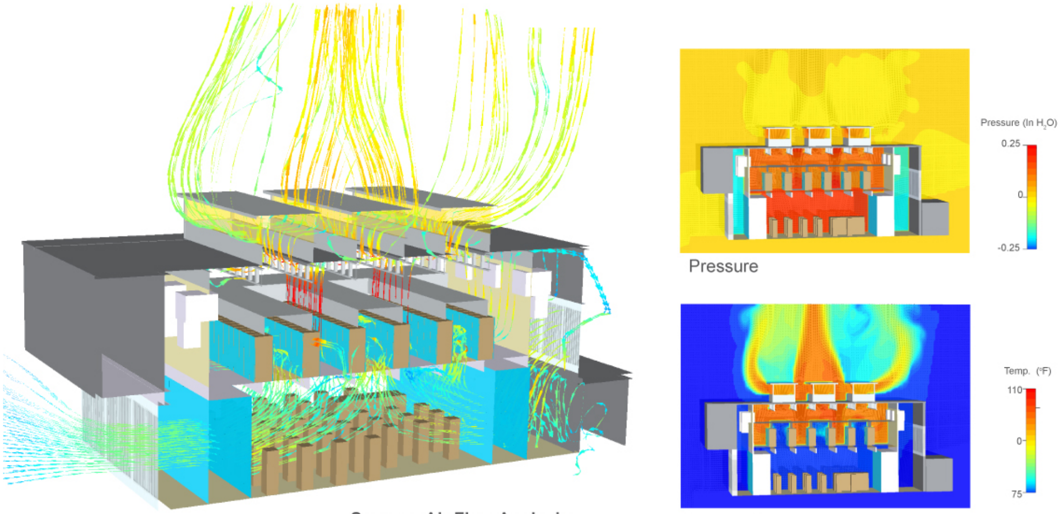


Figure 5. Costs incurred by a single underprovisioned cloud for three options: offloading only to Amazon Web Services (existing data center), offloading to five federated clouds (Open Cirrus 6) and AWS, and offloading to 49 federated clouds (Open Cirrus 50) and AWS.



From "Sustainable IT – a Year in Review", Joyce Dickerson, Nov. 5, 2009 [9]

Normal data center PUE is 2 and above. Exceptional data center PUE is 1,5. Google and similar have a PUE of 1.2. Lower PUE gives a competitive advantage on pricing.

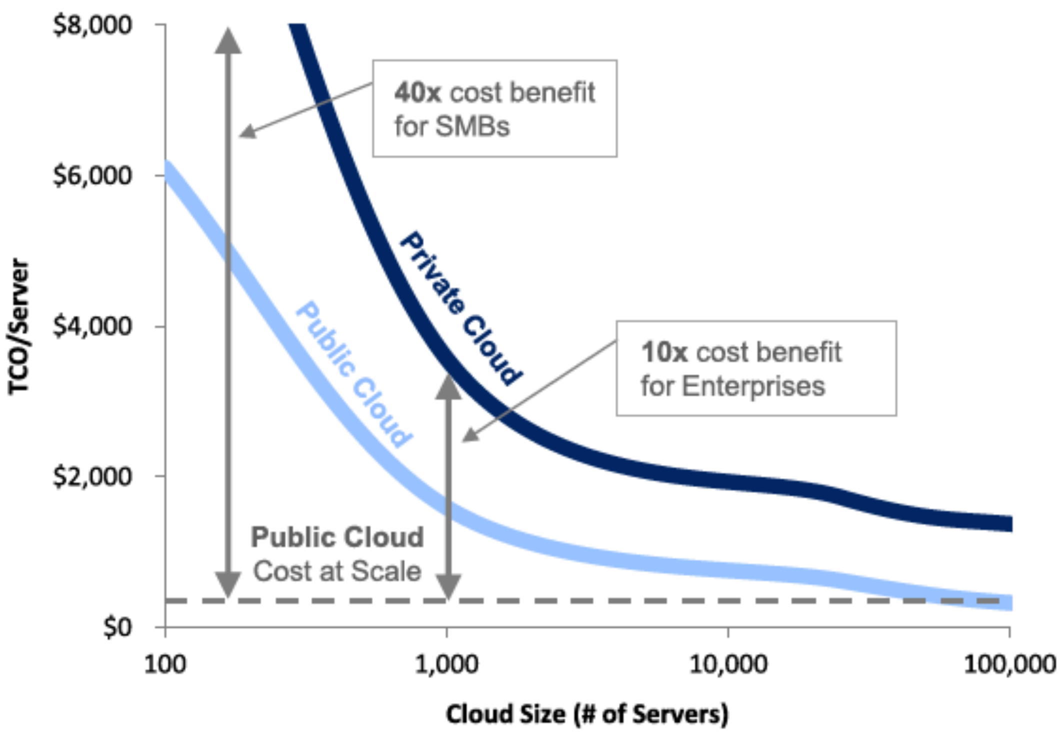
In the overall environmental picture the source of energy need to be taking into consideration, i.e. where the energy is produced and how. In the Nordic region there are a number of alternatives, including thermal alternatives (Iceland), water power plants (Norway, Sweden, Finland).

Cost - public cloud offering

The pilots of the NEON project found that the cost of administration of today's private clouds is too unpredictable, resulting in a overall recommendation towards public clouds. The pricing models of public clouds is highly competitive lead by Amazon, from which we make below comparisons: to deliver the services on internal resources, or to buy it from the (public) cloud.

This brings us back to the TCO results from the most recent report on Cloud and Cost (Nov 2010, Microsoft). As mentioned at the first view of below picture, public cloud offering is overall more cost efficient than the private cloud alternative - for the use cases where both could be used. What was not discussed earlier is the actual functions plotted: how much the user could benefit on lower TCO/server (the y-axis) depending on how many servers (the x-axis) he/she needs.

FIG. 22: COST BENEFIT OF PUBLIC CLOUD



November 2010 calculation from Microsoft [19]

The conclusion from this study is: "--- for organizations with a very small installed base of servers (<100), private clouds are prohibitively expensive compared to public cloud. The only way for these small organizations or departments to share in the benefits of at scale cloud computing is by moving to a public cloud. For large agencies with an installed base of approximately 1,000 servers, private clouds are feasible but come with a significant cost premium of about 10 times the cost of a public cloud for the same unit of service, due to the combined effect of scale, demand diversification and multi-tenancy.

In addition to the increase in TCO, private clouds also require upfront investment to deploy – an investment that must accommodate peak demand requirements. This involves separate budgeting and commitment, increasing risk. Public clouds, on the other hand, can generally be provisioned entirely on a pay-as-you-go basis. "

To make a fair comparison private-cloud vs public cloud - private clouds can be more customizable, but comes with a installation cost and with lower flexibility.

To summarize: we only compare to public cloud in our cost analysis, leaving private cloud alternatives for future studies (see recommendation chapter).

Two things are needed now: to understand the cost of our current data centers, and to get good estimates on what these (or parts of these) services would cost on a public cloud (in this case AWS).

Current cost

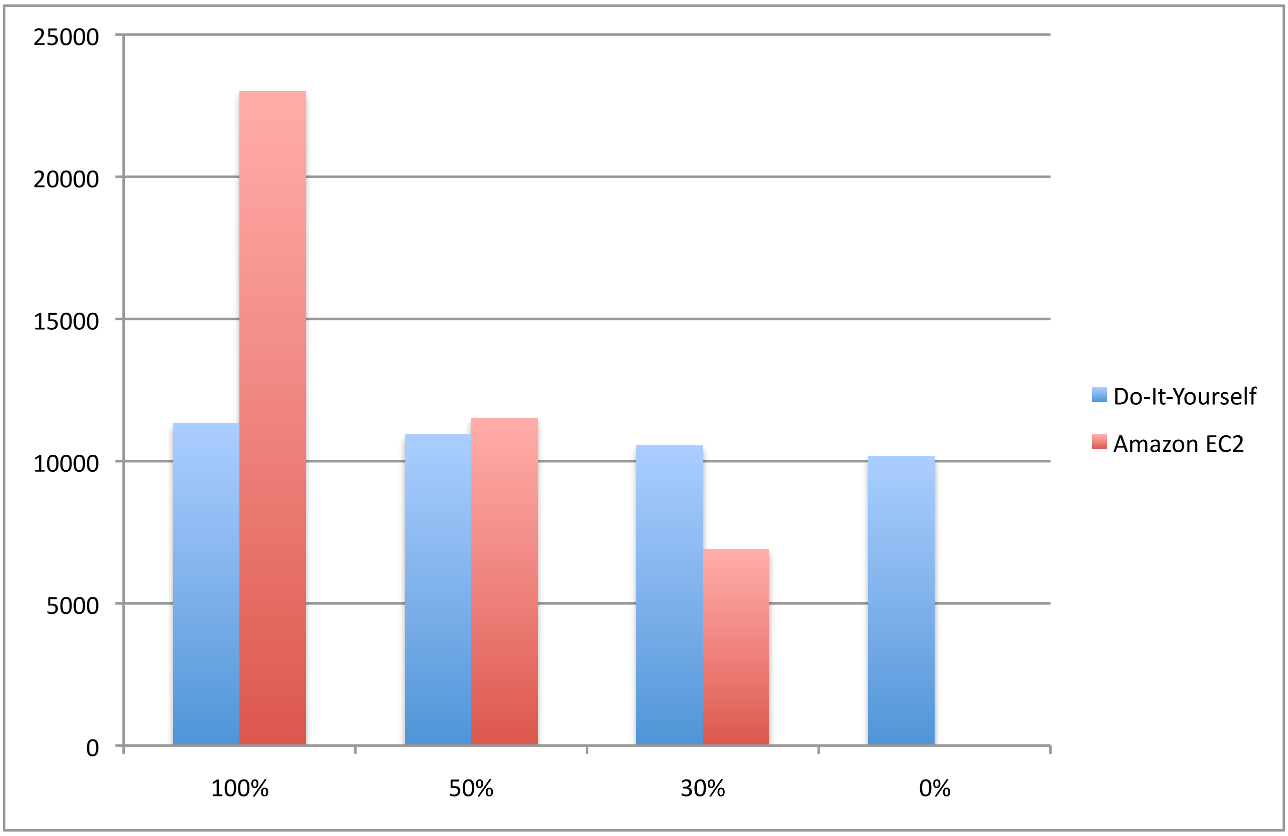
There's a number of studies on this field, and we used approximate numbers from [15], distributed these numbers among the NEON partners for comments.

Amortized Cost	Component	Sub-Components
45%	Servers	CPU, memory, storage systems
25%	Infrastructure	Power distribution and cooling
15%	Power draw	Electrical utility costs
15%	Network	Links, transit, equipment

The overall finding among the NEON partners was: that above numbers are fairly close to their experiences, that most partners don't have their DC costs on that detail, and that especially administation cost was very hard to estimate. The admininistration part of the cost is one area where cloud computing has its benefits - less administration is needed then the actual machines resides on an external site. Still, the difficulty of estimating the current administration cost is due to the administrator's many varying duties, including participation in projects not directly related to the handling of the data centers. Nevertheless: we want to point out this lack of data, and find it most likely that if we could include the actual administration cost of today, cloud services would not make this expense higher (more likely lower).

Specific Cloud Migration Cost Estimates - Norway

One of the NEON partners, Notur, was especially studying public cloud services - with a specific user case (eSysBio) using AWS. During these studies a more detailed cost estimate was done (attached to the report, NOTUR Cloud Cost Estimates - an excel file).



Picture: NOK/node/year versus utility level

Actual number of Notur centers were compared to AWS using the Amazon 'price calculator'[6]. In above example[7] we see a breakeven point at 50% utility level. All examples (above and in Appendix) are Linux instances, based on November 2010 data. The full details are to be found in Appendix VII in the end of this report.

[6] http://calculator.s3.amazonaws.com/calc5.html

[7] High-Memory Double Extra Large - 4 cores nodes - Amazon spot instances pricing

Storage and Network

The factors deciding the storage cost is detailed on:

- Storage/Day/Week/Month
- Reduced Redundancy Storage/Day/Week/Month
- Data Transfer In/Day/Week/Month
- Data Transfer Out/Day/Week/Month
- PUT/COPY/POST/LIST Requests:
- GET and Other Requests

The basic rule for Amazon S3 service of today (November 2010 data) is: 12 euro cents per GB and month - 35 euro cents to get 1 GB up and back with options of direct links to, e.g. Amazon Ireland.

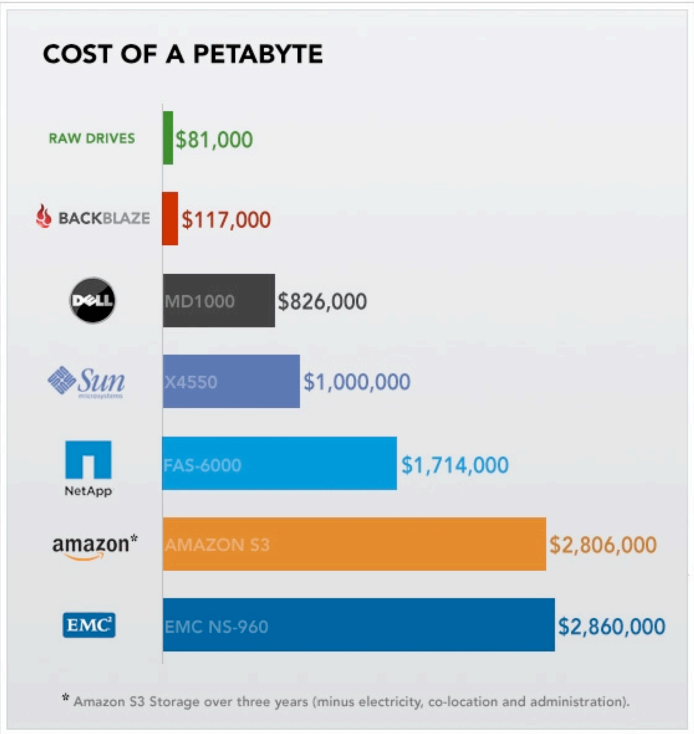


Table of vendor pricing on storage for 3 years[8]

For the upload and download of larger amount of data there are services for direct transport of media storage through e.g. DHL.

8 http://blog.backblaze.com/2009/09/01/petabytes-on-a-budget-how-to-build-cheap-cloud-storage/

Add-on Services

Comparing to Do-It-Yourself, cloud services comes with a number of add-on services. For example from Amazon the list of services is steadily growing, as is from the private cloud vendors. To complete the picture when deciding on alternatives, the user should consider the different use cases

Compute
- **Amazon Elastic Compute Cloud** (EC2)
- **Amazon Elastic MapReduce**
- **Auto Scaling**

Content Delivery
- **Amazon CloudFront**

Database
- **Amazon SimpleDB**
- **Amazon Relational Database Service** (RDS)

E-Commerce
- **Amazon Fulfillment Web Service** (FWS)

Messaging
- **Amazon Simple Queue Service** (SQS)
- **Amazon Simple Notification Service** (SNS)

Monitoring
- **Amazon CloudWatch**

Networking
- **Amazon Route 53**
- **Amazon Virtual Private Cloud** (VPC)
- **Elastic Load Balancing**

Payments & Billing
- **Amazon Flexible Payments Service** (FPS)
- **Amazon DevPay**

Storage
- **Amazon Simple Storage Service** (S3)
- **Amazon Elastic Block Store** (EBS)
- **AWS Import/Export**

Support
- **AWS Premium Support**

Web Traffic
- **Alexa Web Information Service**
- **Alexa Top Sites**

Workforce
- **Amazon Mechanical Turk**

he/she expect to better evaluate the final costs and usability. E.g. load balancing and autoscaling, management tools and most likely to be needed, as is ID management and metering. Cloud is usually look at as a way to lower existing costs, but is also about adding flexibility and new services.

4 Pilots and Lesson Learned

One of the main goal with this project is to report on the state-of-the-art including the HPC for cloud scenario, with pilots and recommendations focusing on non-HPC cloud scenarios. This is also the most urgent issue for today's HPC centers – that non-HPC jobs are taking considerable shares of the runtime on HPC resources.

4.1 How to move non-HPC jobs to a cloud computing environment

This chapter will list the scenarios for moving non-HPC jobs off the supercomputing infrastructure to a cloud-like environment. The scenarios are:

- Existing private cloud solutions in an "Enterprise" (licensed) variant
- Open source private cloud solutions
- Local solutions, completely do-it-yourself
- Public clouds

These scenarios will be described below.

Existing private cloud solutions using licensed software like Enterprise Eucalyptus/ OpenNebula, together with management software such as RightScale.

Though this solution is feature-rich it is also complex and expensive. Extra features are things like quota management, cloud portability and policies on the usage of cloud services per end user. Quota management is often done by setting a virtual price on the local infrastructure and then assigning a budget to a user or a project. Quotas are "soft" quotas, meaning that the user does not get shut down but rather that the administrator gets signaled on reaching a certain usage threshold. Note that via this pricing mechanism one budget can mix and match public and private clouds.

Cloud portability is implemented by adding a meta package system on top of "identical" virtual machines in every cloud. The user then creates a virtual machine by adding packages to the base image in a web based application. Additionally, multiple virtual machines may be bundled in a deployment. Such a deployment can then be launched on a cloud of the user's choosing. Also, administrators may provide rich template virtual machines – we have seen examples from computing instances to Wikimedia services. This concept is often referred to as "IT vending machine".

However, setting up a private cloud has a steep organizational learning curve. Doing this in such a way that it integrates with cloud management software and fully utilizes the added benefits of the management software will add man-years to the initial set up compared with plain old hardware. And as private clouds are rapidly evolving with at east two major releases per year, intensive update cycles will keep this labour-intensive, requiring an organization to acquire new specialistic knowledge of fleeting nature.

We expect as Enterprise private clouds mature more and more the need for separate management software will decrease and the required effort to keep a private cloud running will go down. This will be at least another 18 months though.

Open source alternatives of Eucalyptus/OpenNebula

We have looked in depth at two leading open source private clouds offerings, OpenEucalyptus and OpenNebula. Both solutions are open source, though Eucalyptus also has an Enterprise variant. During the project a third solution became open source (CloudStck) and started rising, specifically in the US realm. We have deployments of OpenEucalyptus and OpenNebula in Finland and Sweden, with the Swedish one (RedCloud) still continuing.

There is a strong tendency towards API compatibility with the Amazon public cloud, recognizing that Amazon currently has set the gold standard. Eucalyptus differs in that it also offers storage, OpenNebula only offers computing at the time of writing.

In using private clouds we have gained a number of insights:

- **Private clouds require above average control of the network topology.** As private clouds effectively provision virtual machines to remote users, the network infrastructure needs to support the cloud model that has been chosen, or vice versa. This requires system and network administrators to closely work together. In practice, this adds significant calendar time in initial set up.

- **Private clouds need "current" hardware.** As the older hardware does not support virtualization we have had e.g. to resolve using Xen's paravirtualization. Xen provides an efficient virtualization platform with low virtualization overhead. Its shortcoming is that it requires modifications to virtual machine's kernel and drivers and is hence difficult or impossible to apply to non Linux operating systems like Microsoft Windows. In other words: private clouds function best with hardware support for virtualization.

- **Private cloud stacks are not as complete as public cloud offerings.** This can be seen quite easily when comparing features of the cloud offerings. Where Amazon[9] offers queueing services, relational database services, load balancers, VPNs to a "private cloud within the public cloud", most private clouds offer only computing. Eucalyptus offers storage as well, but it can be considered sub-par compared to public providers. This implies that significant effort is required to offer a full scale cloud "experience" with private clouds.

Note that the above three points also hold for the Enterprise variants, though the first two points will most likely be solved by the included consultancy in an Enterprise offering. Additional services and middleware (the third point) are often the value-added services of the Enterprise stack, but at significant cost compared to a public cloud.

Finally, review by our peers has revealed that long-running larger private clouds with Eucalyptus were deemed fragile, were OpenNebula was considered more stable, but lacking storage.

---

[9] As Amazon has effectively set the gold standard, we'll use them to compare features etc.

Local solutions

Computational high energy physics (HEP) requires very large, non-standard software installations. This last requirement makes it difficult to use all potentially available resources.

A DIY solution has been implemented on a 24-core farm at CERN: A small library of virtual machine images was put in place and used for high energy physics applications and medical image analysis. For the case of HEP, the images were provided by the CernVM project, for medical image analysis, a minimal Debian image was used.

The main lesson learnt here is that the current state of private clouds is such that for small to medium sized problems, a DIY solution might provide significant benefits in terms of time, money and effort spent while providing an end user solution that is more tailored to the needs of the user base.

Public cloud solutions

Public clouds seem to be the most stable and feature rich offerings – the biggest downside is their price, as some components of the pricing structure appear free for NGIs and NRENs, e.g. bandwidth. The biggest advantages of the public clouds are that they have become so accessible that an average system administrator does not need to build a whole lot of new expertise. It is wise to have system administrators build custom virtual machines on the public cloud to minimize support. Another additional benefit of public clouds is the community around a cloud – for end users and system administrators alike.

Within one domain/brand/flavor of public cloud access management can often be implemented, though not full integrated with existing federations. Quotas are not directly implemented, so monitoring budget is a manual operation unless cloud management software is used. But as soon as one choses one specific public cloud (as is often the outcome from a tender procedure) most other benefits of cloud management software might be too expensive for what they offer.

Finally, public clouds offer instant availability – and thus low turn-around times. This is an excellent feature for organizations starting with cloud computing; it allow for setting up the optimal organizational structure and work on basic cloud knowledge without having to acquire lots of highly specialistic knowledge up front.

4.2 Tests, Pilot Implementations, Gap Analysis

This section will give a brief finding on the tests that have been performed and which tools have been used. We'll also look at the gap analysis with the currently available private and public clouds and their management software on the one hand, and the desired and needed functionality on the other hand.

1. Shortlists public and private cloud infrastructure providers, as well as a shortlist cloud management software. *Rationale: the goal was to have three shortlists that "work well together", so that a) hybrid clouds and b) cloud management becomes realistically testable.*

   Public clouds: Amazon, Rackspace
   Private clouds: OpenEucalyptus, OpenNebula
   Management software: RightScale

   These shortlists reflect the most mature products when the project started in early 2010.

2. Shortlist pilot applications: at least one computationally oriented and at least one storage oriented application. *Rationale: it was expected that besides computationally intensive also storage based applications will benefit from the cloud.*

We have done two pilot applications in Norway on the public cloud:

- ***Cloud backed storage:*** can we offer storage that is cloud backed as if it is a normal disk partition, completely metered, encrypted and elastically scalable. This pilot currently runs successful in the Amazon cloud. See appendix IV for details.

- ***eSysbio pilot:*** can we run part of the eSysbio project in the public cloud. eSysbio aims to develop an e-science environment for supporting systems biology research – and use it to drive Norwegian research within this field. It will conduct research on Web services and service-oriented architecture (SOA) , and use the results to build a collaborative virtual workspace that will facilitate the interdisciplinary exploitation of data, tools and computational resources relevant for systems biology research. See appendix Ifor details.

- ***Cloud deployments:*** At the University of Iceland, a pilot case study has been performed to study issues related to software architecture and Clouds. Support for changes of the run-time and deployment-time architecture has been investigated. In the context of Cloud computing this refers essentially to a deployment. Instead of needing to, e.g., manually fire up new Virtual Machines (VM) and manually start services on them, a scripting language has been applied to automate deployment of applications in the Cloud.

- ***Private cloud infrastructure pilots:*** what is the current state of private clouds and can we run pilots on them? To this end we have set up OpenEucalyptus and OpenNebula in Sweden and Finland. Based on the maturity level of these products we deemed it impossible to do a user-based pilot in those environments within the scope and manpower of the NEON project. See appendices II and V.

3. Chose and set up a public cloud infrastructure for a pilot application. *Note: the goals was to do this on local sites and will then be made to work together with other sites and the management software (see point 6).*

   Public clouds have been set up by Norway and Iceland, see appendices I and III

4. Chose and set up a private cloud infrastructure for a pilot application. *Note: extra attention will be given to multiple-site management.*

   This has been done in Finland and Sweden; Sweden has also set up a private cloud with multiple availability zones, see appendix II.

5. Chose and set up management software for cloud infrastructure. *Rationale: the management software will be crucial in provisioning, metering and in migration between clouds (hybrid clouds).*

   We have set up accounts at RightScale to test the management infrastructure. Though RightScale's feature set is impressive and covers most needs for (very) large deployments, discussions about the cost of Rightscale deemed it too expensive compared to spending on the rest of the cloud infrastructure, especially in an initial phase. As part of a large scale tender process Rightscale and similar services could be expected to play a role in offerings with private/public clouds.

   Among the features of RightScale are access management, quota management, a cloud component repository, attaching a virtual price on private cloud usage and a system for setting up virtual images in a cloud-independent manner.

6. Feasibility of multi-domain support (availability zones) for private clouds. *Rationale: one could consider a NGI as an "availability zone" - here we have learnt whether this is a useful comparison and how to implement cross-domain private clouds using the management software.*

   This has been implemented by Sweden with two sites locally, and a possible third German site joining later. Although it is certainly doable, tight coupling with network provisioning that will differ across borders make this impracticable. This is due to the fact that as soon as the network provisioning has been chosen on the main site, all other sites must follow the same model. If network provisioning is the same, adding multiple zones is straightforward though. Thus feasibility of multiple domains is largely organizational.

7. Gap analysis on cloud offerings versus needs (qualitatively) in user base, based on pilot experiences. *Rationale: when doing pilots, user feedback will give a qualitatively oriented feedback on using HPC in a cloud for private and public clouds, as well as the management software. This guarantees that cost is not the only metric.*

   Based on the eSysbio pilot, the cloud backed storage pilot, and the pilots that have set up private cloud infrastructures we have concluded that the current public cloud offerings are superior to private cloud offerings.

   As needs we simply consider the mimicking the current state of affairs on HPC computers, i.e. providing a similar user experience in a transparent way. The gaps then are:

   Gaps private clouds:
   - *No mature storage offering* - storage offerings are either no part of private clouds, or perform sub-par
   - *No other middleware offerings out of the box* - services like databases, queues etc. must be deployed and maintained by the providing site.
   - *Heavy reliance on system and network administrator expertise for end users* - essentially the downside for users when having full control
   - *Separate management software* - either you pay for an Enterprise version, or you pay for separate management software/services.
   - *Identity management integration limited to users uploading their own certificates* - see also next point (point 8).
   - *Rapid update cycle of core infrastructure software*
   - *Quota management hard or non-existent*

   Gaps public clouds:
   - *More reliance on system administrator expertise for end users* - same downside as above, though network knowledge might be less of an issue.
   - *Separate management software* - this depends on the particular cloud solution and the level of management required. Obviously, public cloud offering offer some level of management if only for billing purposes. Public clouds develop themselves rapidly in this area, but not in a compatible way.
   - *Identity management integration limited to users uploading their own certificates* - see also next point (point 8)
   - *Quota management hard or non-existent* - this is obviously not in the interest of public cloud providers. Why limit usage if that's what they sell?

   On the plus side, the pilots have learnt us that the immediate availability speeds up project set up significantly. Also, the specialistic expertise at public cloud providers seems to be such that e.g. hardening of the infrastructure might be better than you could do it yourself. This is due to economy of scale.

8. Gap analysis on integration with existing AAI for public and private cloud offerings.

Rationale: AAIs are well deployed, both locally and internationally (eduGAIN). Utilizing federated identity management should greatly foster adoption by easing access.

This has turned out not to be an option. The best both public and private clouds currently do is allowing users to upload their own client certificates for starting/stopping services. This is after they have signed upp in a custom identity management system. These systems tend to be closely coupled with the provisioning infrastructure, rendering AAI integration nearly impossible without a significant update from private and public cloud providers alike.

9. Gap analysis on integration with existing metering infrastructures. *Rationale: do the current metering infrastructures have what it takes and how do they compare to the metering built into clouds and cloud management software?*

Current cloud solutions provide APIs for monitoring usage, but do not offer quota management or cost control. Third-party management systems do provide soft quotas. Currently there exists a wide gap with respect to quota management and cost control for every cloud provider. Cloud management software bridges this gap, but adds an *extra* management infrastructure. We expect that the gap will close from three sides in the 2011-2012 timeframe:

1 - commercial cloud management software will be able to manage "bare metal" more and more
2 - cloud solutions will slowly add cost control and quota management features
3 - open source cloud management solutions will thrive and add quota management and cost control

10. Architecture and cost analysis for integration with existing AAI infrastructure. *Rationale: once we know the gaps with the current AAIs the questions will be a) can we integrate b) if so, how? c) at what cost?*

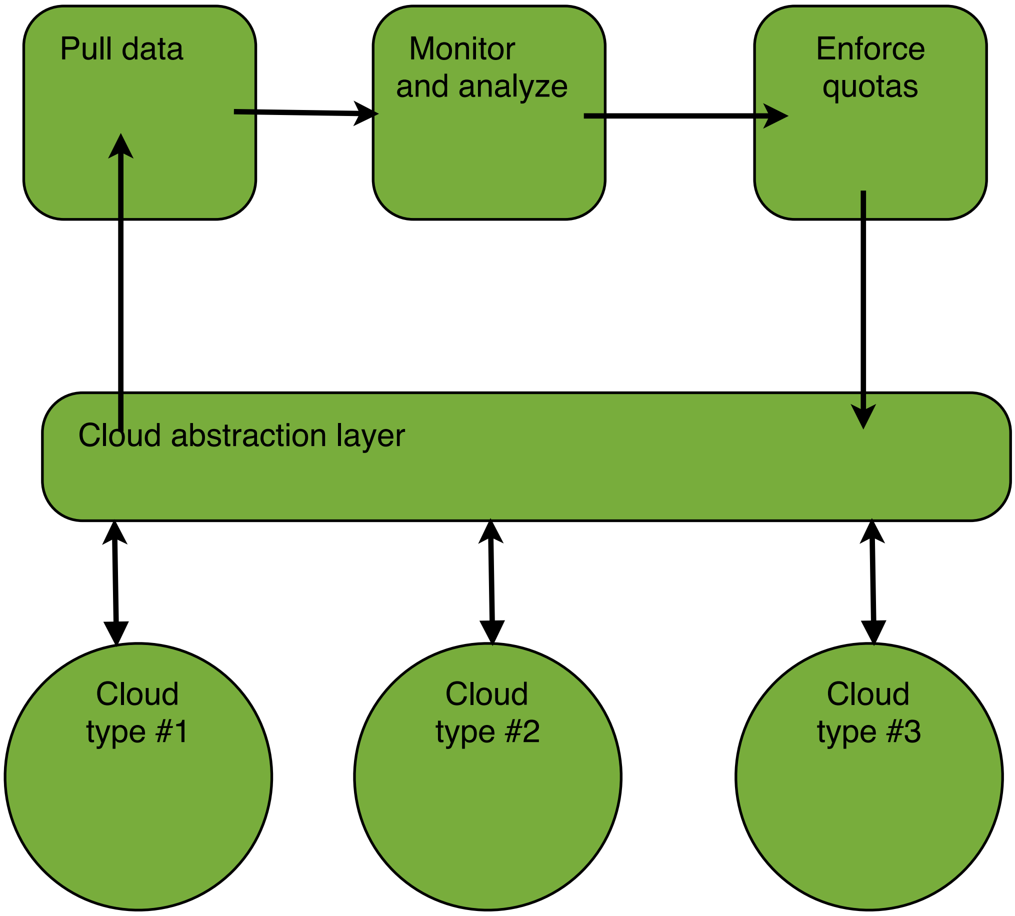
This is not an option, further cost analysis on this point has proven useless until public and private cloud services enhance their identify management. It would result in an large rewrite if e.g. private cloud software or DIY solutions as per the Danish pilot, see Appendix VI

11. Architecture and cost analysis for integration with existing metering infrastructure. *Rationale: see previous point.*

As clouds currently have no cost control only monitoring at best, there are multiple options.

The first option is using a (hybrid) cloud management solution such as Rightscale. These solutions provide charging of private clouds infrastructures as well, so costs for a project can span multiple cloud types. Also, these solutions provide soft quotas (signaling) which gives a weak kind of cost control.

A second option would be to use the APIs provided by the cloud providers to continuously polling for usage information and then implementing a quota system yourself. The cost of developing such a system is believed to be $50,000-$100,000 and the functionality will be basic. Maintenance (given the rapid cloud infrastructure developments) will be at least another $50,000 on a yearly basis.



Another option might be donating to an open source cloud management project. This will largely eliminate maintenance costs. The cost should be no higher than $50,000 given the nature of open source projects.

Finally, simply waiting while the market or open source community solves these problems is a valid option as well. In the meantime, 0.1 FTE should be enough to monitor the cloud cost on a daily basis manually.

5 Dissemination and Outreach

NEON project have been active in presenting its ongoing activity internationally. Main purpose for this has been feedback from other similar projects, and education of users. Through these activities NEON found its pilot users, eSysBio.

The Norwegian metacenter for computational science

meta

The Cloud

I recently heard a podcast where the speaker made an interesting case. He argued that if you want to do something, in general the safe way to do it is often to do the *right* thing for the *wrong* reasons. He argued that a lot of systems are organized that way, e.g. we breathe oxygen not because we need oxygen, but because the levels of carbon dioxide in our blood are too high. We don't lock away criminals because they are criminal, but because their income tax doesn't match. And so on.

AUTHOR
Maarten Koopmans,
maarten@vrijheid.net

Granted, this is a peculiar way to start an article about cloud computing, but with a hyped subject such as cloud computing it is often helpful to remind yourself that you (or many like you, such as the people hyping the subject) *may* have started by doing the right thing for the wrong reasons. As long as you remind yourself as well what the right reasons ought to be and you closely monitor your overall progress, you should be fine. So it is perfectly OK to work on cloud computing because it is more cost effective, provides more end user flexibility, is cool technology (this helps getting good people), all because of....

More science
Let's take a look at what cloud computing is and try to define it. Cloud computing has many definitions but I think the *best* use case for cloud computing is defined by the term "utility computing". Much like utilities such as gas, electricity, water etc. the term "utility computing" comprises computing on-demand, always available in sufficient quantity, and paid for the actual usage (dubbed pay-as-you-go). The difference with the above mentioned utilities is that cloud computing – and cloud storage – has different service parameters that you can chose from, e.g.

20

NEON cloud chapter in Meta - Number 2, 2010[10]

5.1 List of Dissemination and Outreach Activities

2010.03.05 Terena TF-Storage, Utrecht: Northern European Cloud project, http://www.terena.org/activities/tf-storage/ws7/slides/050310-storage5-neon-maarten.pdf

2010.03.08, Technion, Haifa & TAU, Tel Aviv, Israel: Cloud computing and Innovative Companies

2010.03.16, OGF Europe, Munich: ECEE - Enabling Clouds for EsciencE (BOF)

2010.03.24, PDC Seminar, Stockholm: Cloud Computing Security - an Oxymoron?

2010.03.28-29, Inforte Seminar, Helsinki: Cloud Computing and Service Engineering (Cloud Security & Cloud and Innovative Companies)

2010.04.13, NDGF AHM, Uppsala: Northern Europe Cloud Initiative Status Report

2010.05.04, Nordugrid 2010, Ljubljana: NEON and VENUS-C - two new cloud projects

2010.05.19, NEON Workshop, NOTUR 2010, Bergen: x10 Questions and Answers about Cloud Computing

2010.05.31 TNC 2010, Vilnius: Clouds from the trenches, http://tnc2010.terena.org/core/getfile.php?file_id=127

2010.09.16, EGI TF, Amsterdam: ECEE Workshop (Chair)

2010.10.06, Mindtrek, Tampere: What are the services Cloud could offer? (presentation here)

2010.10.22, Cloud Computing, CSC, Espoo: Cloud Innovation Platforms (presentation here) (youtube)

2010.11.16, Danish Research Network, Middlefart: Cloud Computing for eScience - today and future (presentation here)

---

10 http://www.notur.no/publications/magazine/pdf/meta_2010_2.pdf

5.2 Impact on International Level

NEON has been active in a number of international collaborations. As coordinator of the ECEE[11] - Enabling Clouds for EsciencE - collaboration, NEON shares a roadmap with 9 other projects in EU. During the NEON project ECEE met twice for longer workshops at OGF and EGI TF.

NEON coordinator is now also part of the SIENA[12] Roadmap Editorial Board, led by Martin Walker - defining recommendations on distributed computing for the European Parliament.

6 References

[1] "Open Cirrus Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research", Roy Campbell, Indranil Gupta, Michael Heath, Steven Y. Ko, Michael Kozuch, Marcel Kunze, Thomas Kwan, Kevin Lai, Hing Yan Lee, Martha Lyons, Dejan Milojicic, David O'Hallaron, and Yeng Chai Soh

[2] "The Cost of a Cloud: Research Problems in Data Center Networks", Albert Greenberg, James Hamilton, David A. Maltz, Parveen Patel

[3] "Do Clouds Compute? A Framework for Estimating the Value of Cloud Computing", Markus Klems, Jens Nimis, and Stefan Tai

[4] "(MC2)2: A Generic Decision-Making Framework and its Application to Cloud Computing", Michael Menzel, Marten Schönherr, Jens Nimis, Stefan Tai

[5] "Towards a Model for Cloud Computing Cost Estimation with Reserved Instances", Georg Singer, Ilja Livenson, Marlon Dumas, Satish N. Srirama, and Ulrich Norbisrath, 2010

[6] "The Economics of the AWS Cloud vs. Owned IT Infrastructure", Amazon, 2009

[7] Jacko Koster, private communication

[8] "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS", Ali Khajeh-Hosseini, David Greenwood, Ian Sommerville

[9] "Sustainable IT – a Year in Review", Joyce Dickerson, Nov. 5, 2009

[10] "HPC benchmarks on Amazon EC2". S. Akioka, Y. Myraoka, 2010

[11] "The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise", Ali Khajeh-Hosseini, David Greenwood, James W. Smith, Ian Sommerville

[12] "A Simple Model for Determining True Total Cost of Ownership for Data Centers", J. Koomey, K. Brill, P. Turner, J. Stanley, B. Taylor

[13] "Advantages of a Dynamic Infrastructure: A Closer Look at Private Cloud TCO, S. Bain, I. Read, J. Thomas, F. Merchant, IBM SWG Competitive Project Office, April 2009

[14] "The Cost of Doing Science on the Cloud: The Montage Example", E. Deelman, G. Singh, M. Livny, B. Berriman, J. Good

[15] "Above the Clouds: A Berkeley View of Cloud Computing", M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, Feb. 2009

[16] "Clearing the air on cloud computing", McKinsey & Company, March 2009

---

11 http://www.scientific-cloud.org/

12 http://www.sienainitiative.eu/

[17] “Evaluating the Cost-Benefit of Using Cloud Computing to Extend the Capacity of Clusters”, M. Assuncao, A. de Costanzo, R. Buyya, June 2009
[18] “Cost-Benefit Analysis of Cloud Computing versus Desktop Grids”, D. Kondo, B. Javadi, P. Malecot, F. Cappello, D. Anderson
[19] “The Economics of the Cloud”, Rolf Harms (rolfh@microsoft.com) and Michael Yamartino (michael.yamartino@microsoft.com)

7 Appendix

Below we have added a list of short reports documenting the findings from the various hands-on testing being made during the NEON project. These rapports are included in free-form as addendums to the document.

I eSysbio: a bio-informatics pilot on the public cloud, UNINETT Sigma

II NEON Report for Eucalyptus and OpenNebula experiences and lessons, PDC-HPC, KTH

III Cloud pilot case study performed at the University of Iceland

IV Cloud backed storage, UNINETT Sigma

V Cloud experiments UH-HIP

VI A pilot virtualized batch facility, grid.dk

VII NOTUR Cloud Cost Estimates

7.1 *Appendix 1- eSysbio - a bio-informatics pilot in the public cloud - Norway*

Author: UNINETT Sigma

Introduction

A large fraction of today's eScience users run single node jobs. Many of these single node jobs may run equally well on a commodity machine and with a smaller effort from the user. The aim of the NEON project is to review these promises and summarize the overall offering cloud computing could give to the Nordic eScience community. The project includes pilot tests to get real performance and cost data as well as real hands-on user experience – all to prepare a report for guidance to the NDGF SB its stakeholders, collaborators and users.

The eSysbio project is executed by UniBCCS and aims to develop an e-science environment for supporting systems biology research – and use it to drive Norwegian research within this field.

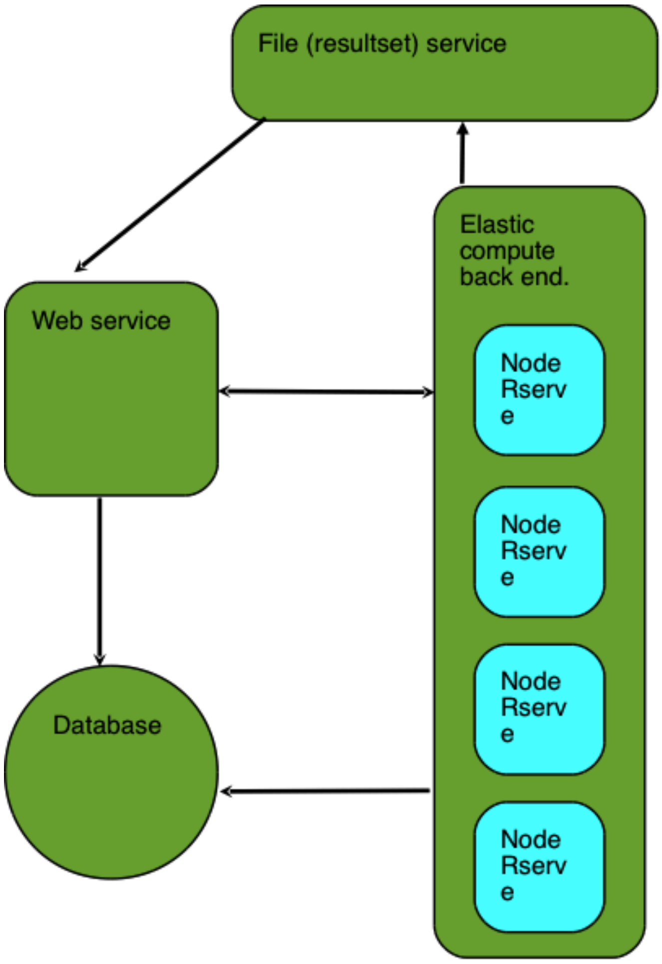
Goals

Offer part of the eSysbio service with an elastic computing back end. A part of the eSysbio service will be offered using an elastic computing back end (see "Technological background") to provide the illusion of unlimited computing capacity for long running jobs. This part of the eSysbio service will thus provide end users with a service that will handle peak performance without introducing scheduling batch jobs.

Assess feasibility for a structural offering. The feasibility in terms of user experience both for the eSysbio project (the programmer as user) and the actual end user will be assessed qualitatively.

Technological background.

As part of the eSysbio project different web services are and will be developed for running compute intensive bioinformatic services or applications. These computations typically runs on a server, and the web service use some sort of polling scheme the keep track of the jobs. Running these type of jobs could fit well into the cloud paradigm. Since the computing activities may vary in terms of intensity of usage, an elastic back end looks an ideal candidate for maintaining performance in an effective way. Schematically, this would look like:



I.e. the elastic compute back end would scale depending on load and performance parameters between a minimum and maximum value. Each compute node registers and deregisters in the web service's database and "pings" to keep the reference fresh. When a job plus data is submitted, a node is assigned by the web service based on the database with availability information. On completion, the result set is transferred to a separate service to prevent data loss when terminating part of the elastic compute back end in scaling down.

Results

The eSysbio pilto was up and runnig fairly quickly (4 mandays of work) on the computational side. On the storage side we have chosen to test drive the other pilot infrastructure, cloud backed storage, for storing e.g. results after a computation. This has added some calendar time due to testing and being the first to truely integrate this service, but it has been successfully done.

To asses feasibility we have gone thourgh the questions below - the answers make clear tat there is added value in a public cloud service and how one should organize it:

How does public cloud computing compare to traditional computing, what are the pros/cons?
There have been no big differences.

Pros:
- freedom
- fast startup
- mixing and matching different types of VMs.

Cons:
- You need to have access to basic admin skills.

How do you want to have access to cloud computing resources?
In this pilot we have used the API and the command line. The ID and secret key provided by Amazon was good enough. The secret key protection was actually an improvement over e.g. grid certificates. Certificates are harder to understand and acquire for non-physics/grid users. Identity management seems not the biggest priority - i.e. it is not a problem to have yet anoher set of credentials.

What level of end user support would be required (think custom Linux images).
Base linux images with default availbility of the other services (cloud backed storage, AWS command line tools) are necessary and will drive support effort down. Up to date Linux distros like Ubuntu server for the EC2 cloud will help greatly.

How should support be organized?

- Wiki pages
- tutorials
- workshops
- mailing list

Have you been able to use "common" communities and if so, to what degree?
Common communites, like the ones from Amazon, have not been used. In stead the local support from UNINETT SIgma was called upon. This underlines the importance of local support presence.

Should a "local" community be organized, if so, how?
Yes, with a strong local "support", monitored by someone who has e.g. 0.3-0.5 FTE (estimate from pilot user). This could also be done by local sysadmins on-site from NOTUR.

Experience with cloud backed storage, would you like to see it as a service? if So, what do you think is important there?
Absolutely. An easy- to use storage service is necessary for storing e.g. computing results. Data sharing is a key feature that needs to be added. Adding other users by "power" users would be nice as well.

7.2 Appendix II - Eucalyptus and OpenNebula - Sweden

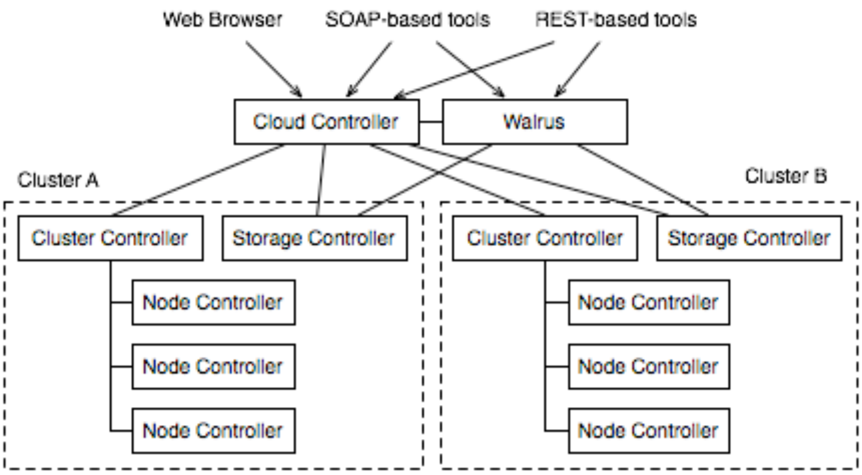
Author: Zeeshan Ali Shah, PDC - HPC , KTH

In this report I will describe the setup, problems faced and solution for Eucalyptus and Opennebula . In first part we will write for Eucalyptus and in 2nd section we will present Opennebula in detail.

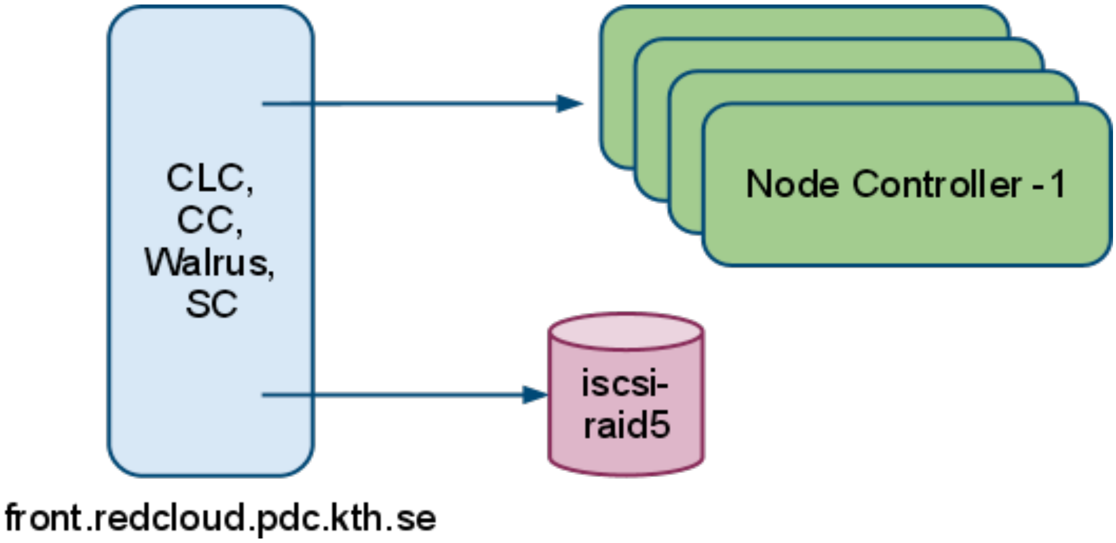
Ecualyptus

Eucalyptus is software that implements scalable IaaS-style private and hybrid clouds. The Eucalyptus architecture is highly modular with internal components consisting of Web services, which make them easy to replace and expand. Eucalyptus' flexibility enables it to export a variety of APIs towards users via client tools. Currently Eucalyptus implements the Amazon Web Service (AWS) API, which allows interoperability with existing AWS-compatible services and tools. This also allows Eucalyptus users to group resources drawn both from an internal private cloud and external public clouds to form a hybrid cloud. [1]
Architecture:

In NEON setup we used the eucalyptus for offering Private Cloud IaaS . The standard architecture consists on :



In PDC the current setup is :



System Configuration :

Front End :
Ubuntu 10.04 Server 64 bit
2 CPUs one core each
Eucalyptus 1.2.6
Nodes :
Centos 5.4 64 bit
2 CPUs one core each
Eucalyptus 2.0
XEN Virtualisation
Networking mode:
Static mode of networking used from Pool of Ips and Mac addresses ,
This required to run DHCP on front-end

Problems , Issues and their Solution :

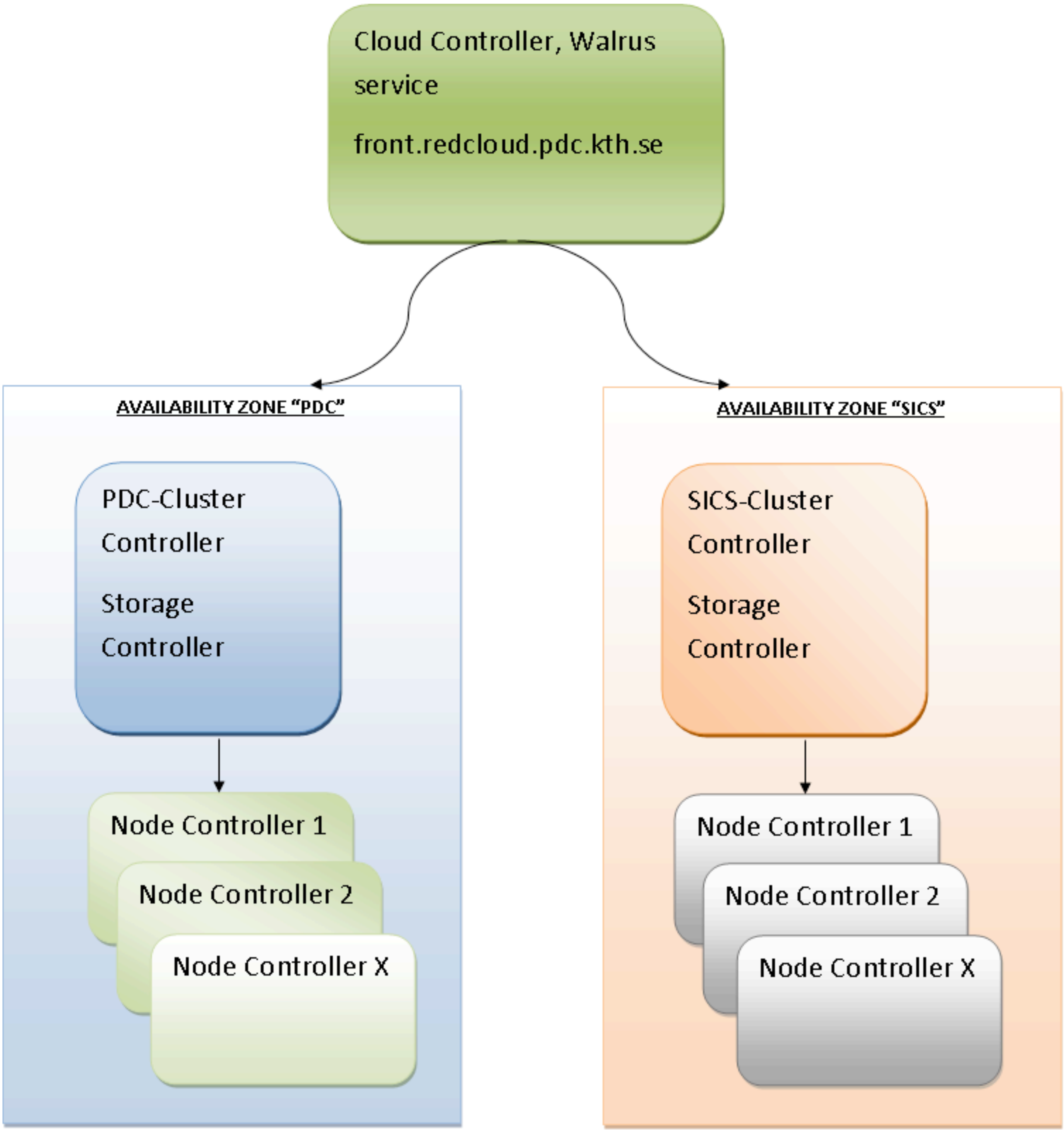
With out HT enabled cpu : As on our hardware HT was not possible which means we cannot KVM as virtualization . But Ubuntu on other side has closed their support for XEN , so initially we faced issues in Installation Ubunutu Enterprise cloud from CD .
We solve this by running CENTOS 5.4 on nodes and install XEN on it. yes it requires more work but we had no option with the existing hardware .
IPs pool issue:
It was the most tricky part as we need IP address from Networking team of our department. We applied for pool of IP address which took around 7-8 weeks . This is due to the reason that our network team made a segment specially for cloud . (due to myth of cloud security) .
From System configuration as you can see that , we have Ubuntu on front end and Centos on nodes which end up doing manual configurations like copying ssh keygen transfer from front to nodes etc. Which actually can be automatically configure by Ubuntu Enterprise Cloud.
Eucalyptus Image:
Initially i have placed an eucalyptus image into repository and it was rather quick and straight forward.

Multiple Availability Zones:

In this report we mentioned the technical overview about cross-domain availability zones between PDC-HPC (PDC Center for High Performance Computing) and SICS (Swedish Institute of Computer Science).
Landscape View :


Swedish Institute of Computer Science
SICS
Kista
Edsviken
Rådan
Danderyd
Ekebyparken
Djursholm
Stadsparken
Granparken
Idrottsparken
Ulriksdals slottsträdgård
Igelbäcken
Lilla Ursvik
Stocksund
Kungshamra
Bergshamravägen
Enköpingsvägen
Frösunda
Råstasjön
Lötsjön
Filmstaden
Råsunda
Brunnsviken
Hagaparken
Haga
Stockholms universitet
Mariehäll
Sundbyberg
Ballstavägen
Lilla Alby
Solna
Skytteholm
Stockholm-Bromma flygplats
Karolinska Sjukhuset
Vanadislunden
KTH
VETENSKAP OCH KONST
PDC Center for High Performance Computing

Architecture View :



SHELL View

From EUCA command Shell you can view the Above Availability Zones like this .

```
zashah@napoletana:~/.euca$ euca-describe-availability-zones
AVAILABILITYZONE    pdc     130.237.221.231
AVAILABILITYZONE    sics    193.10.64.86
In Verbose Mode:
zashah@napoletana:~/.euca$ euca-describe-availability-zones verbose
AVAILABILITYZONE    pdc     130.237.221.231
AVAILABILITYZONE        |- vm types     free / max   cpu   ram   disk
AVAILABILITYZONE        |- m1.small     0012 / 0014   1    192    5
AVAILABILITYZONE        |- c1.medium    0012 / 0014   1    256    5
AVAILABILITYZONE        |- m1.large     0006 / 0007   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0006 / 0007   2   1024    20
```

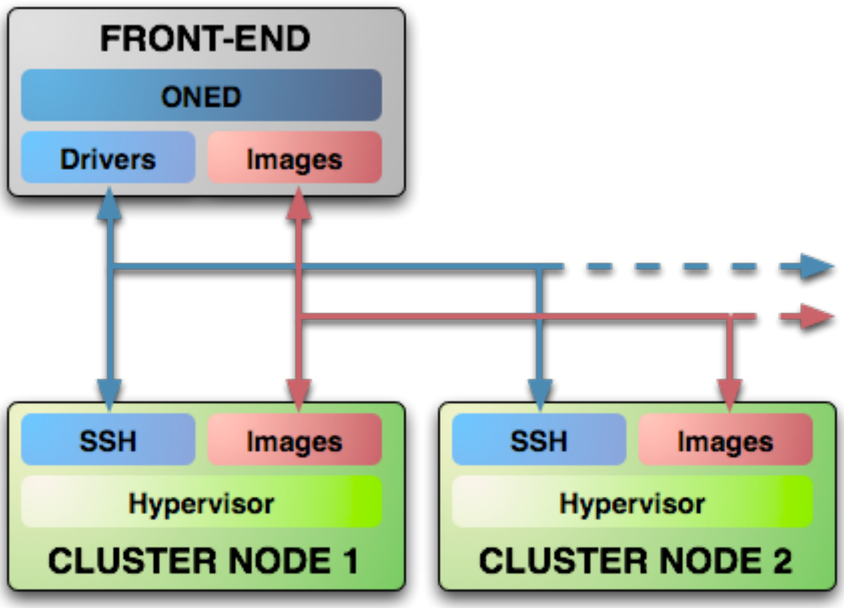
AVAILABILITYZONE		|- c1.xlarge	0000 / 0000	4	2048	20
AVAILABILITYZONE	sics	193.10.64.86				
AVAILABILITYZONE		|- vm types	free / max	cpu	ram	disk
AVAILABILITYZONE		|- m1.small	0036 / 0036	1	192	5
AVAILABILITYZONE		|- c1.medium	0036 / 0036	1	256	5
AVAILABILITYZONE		|- m1.large	0018 / 0018	2	512	10
AVAILABILITYZONE		|- m1.xlarge	0018 / 0018	2	1024	20
AVAILABILITYZONE		|- c1.xlarge	0009 / 0009	4	2048	20

Storage Controllers:
euca_conf --list-scs
registered storage controllers:
sics 193.10.64.86
pdc 130.237.221.231
Cluster Controllers:
euca_conf --list-clusters
registered clusters:
sics 193.10.64.86
pdc 130.237.221.231
Selection of ZONES in creating Instances
euca-run-instances **-z** sics ……………………
euca-run-instances **-z** pdc ………………………..

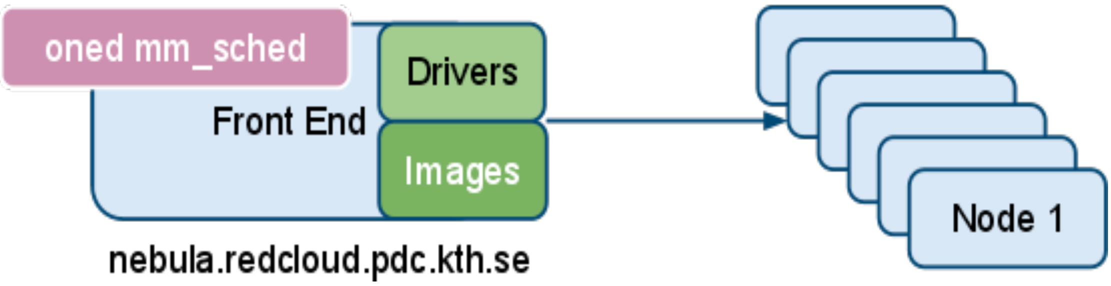
Open Nebula Cloud Platform

OpenNebula is a fully open-source toolkit to build any type of IaaS cloud: **private**, **public** and **hybrid**. [2]

Architecture:
Standard Architecture:



PDC Architecture:



System Configuration :
Front End :
Centos 5.4 Server 64 bit
2 CPUs one core each
Open Nebula : Oned and mm_sched
Nodes :
Centos 5.4 64 bit
2 CPUs one core each
SSH
XEN Virtualisation
Networking mode:
Static mode of networking used from Pool of Ips and Mac addresses ,
This required to run DHCP on front-end

Problem Issues and Solution:
Storage solution :
This is one of the draw back for opennebula as uptil now there is no Storage Controller and Storage service like Walrus (Eucalyptus)
So we end up using either Shared FS (e.g NFS) or SSH transfer
In PDC we adapted both approach initially but later on moved to ssh for eas of use and cross border cluster networking . As it is not secure to run NFS on internet
Networking :
This went very easy as we used from the pool of Ips we got from Eucalytpus installation .
You can specify IPs and Mac in template or have specify any network option maintain by Opennebula and which can be manage by ***onevnet*** command
Images:
Eucalyptus images cannot run out of the box in Open nebula , we used by raw disk image of xen and create a sample template for opennebula .

Personal feedback from Learning

Following are personal feedback from my experiences during installation of both system.
Eucalyptus is more for EC2 , AWS compliant solution , since their reference model is AWS which makes very easy for them to enhance their platform . Installation is easy. Easy to form availability zone
However , walrus service may be overloaded due to frequent and high usage . Platform Not easy to extend
Opennebula , is like another Virtualization management solution plus some more things easy to install, very light weight, easy to extend with several plugins, availability zone multiple (have not tried) .

Reference

http://open.eucalyptus.com/learn
http://opennebula.org/about:technology

7.3 Appendix III - Cloud pilot case study - Iceland

Author: University of Iceland

1. Overview

At the University of Iceland, a pilot case study has been performed to study issues related to software architecture and Clouds. Support for changes of the run-time and deployment-time architecture has been investigated. In the context of Cloud computing this refers essentially to a deployment. Instead of needing to, e.g., manually fire up new Virtual Machines (VM) and manually start services on them, a scripting language has been applied to automate deployment of applications in the Cloud.

To this aim, the Architectural Scripting Language (ASL) that has been developed earlier (Ingstrup, Hansen, 2009) has been adapted to the context of Cloud computing. For example, ASL considers entities such as component (a unit of deployment), interfaces, or devices (in the context of Cloud computing: a node or VM). On these entities, operations such as creating a node or starting a service on a node are supported. ASL allows automating these operations, e.g. using a script shown below (note that as part of the pilot, only applications that are based on the dynamic module system OSGi are supported):

```
// create a large instance referenced as "server"
create_instance_device("large", "server")
// lets start the device, e.g . turn the computer on.
start_device ("server")
// next, we install our components on the server
install_component("userinterface", "http://URI/ui.jar" , "server")
install_component("messengerlogic" , "http://URI/mess.jar", "server")
install_component("dataconnector", "http://URI/db.jar" , "server")
//now, all the components have been installed on the device , we
can start them one by one
start_component("userinterface", "server")
start_component("messengerlogic", "server")
start_component("dataconnector", "server")
```

Automating these operations (i.e. automated deployment and runtime architecture evolution) safes a lot of work and is in fact a logical step subsequent to automated built. In contrast to other Cloud platform-specific scripts, ASL-based scripts have the advantage of being Cloud platform-independent and thus being re-usable in different environments, e.g. when changing the Cloud provider.

2. Pilot case

For evaluating applicability of ASL for the Cloud, it has first been tried to set-up a private cloud at the University's computing center. However, this failed due to lack of human resources at the University's computing center. Hence, the public Cloud Amazon EC2 has been used instead. While the concepts and the syntax of ASL are platform-independent, its implementation is platform specific. Hence, the ASL operations have been mapped to automated EC2 operations (However, due to these different abstraction levels ASL scripts can be re-used on completely different (Cloud) platforms, only the ASL implementation needs to be adapted).

As a case study, an existing Java-based raytracing/rendering application ("Sunflow") has been modified to support distributed processing. ASL has been used for deployment and runtime architecture evolution (i.e. adding VMs). The applicability of ASL for Cloud (Amazon EC2) applications has been evaluated. As a result, ASL proved suitable for the studied dynamic Cloud application. A performance evaluation reveals that ASL is slightly slower than a hard-coded Groovy script that makes hard-coded calls via the Amazon AWS DSK.

3. References

Ingstrup, Hansen: "Modeling architectural change: Architectural scripting and its applications to reconfiguration", Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) 2009 and European Conference on Software Architecture (ECSA) 2009, IEEE, 2009

7.4 *Appendix IV Cloud backed storage for end users - Norway*

Author: UNINETT Sigma

1. Introduction

This document describes the pilot of a cloud based storage solution for end users. Private and public clouds are becoming readily accessible. Their storage interface is purely HTTP REST or SOAP based and requires a significant effort of the application designer and implementer, rendering the benefits of a cloud-based public or private storage mechanism useless for average end users.

To support average end users the technical architecture of a cloud "drive" without the need for a special client will be implemented. This will allow end users to "map" the cloud as a network drive from their computers and devices, and share their data. End users using computing clouds can use the same interface to access data from both computing clouds and their own computers in a familiar way.

The goal is to learn what the added value is of services that make cloud components easier accessible *and* to get first hand feeling for the effort required to use clouds in such a way (to be better able to guide our end users).

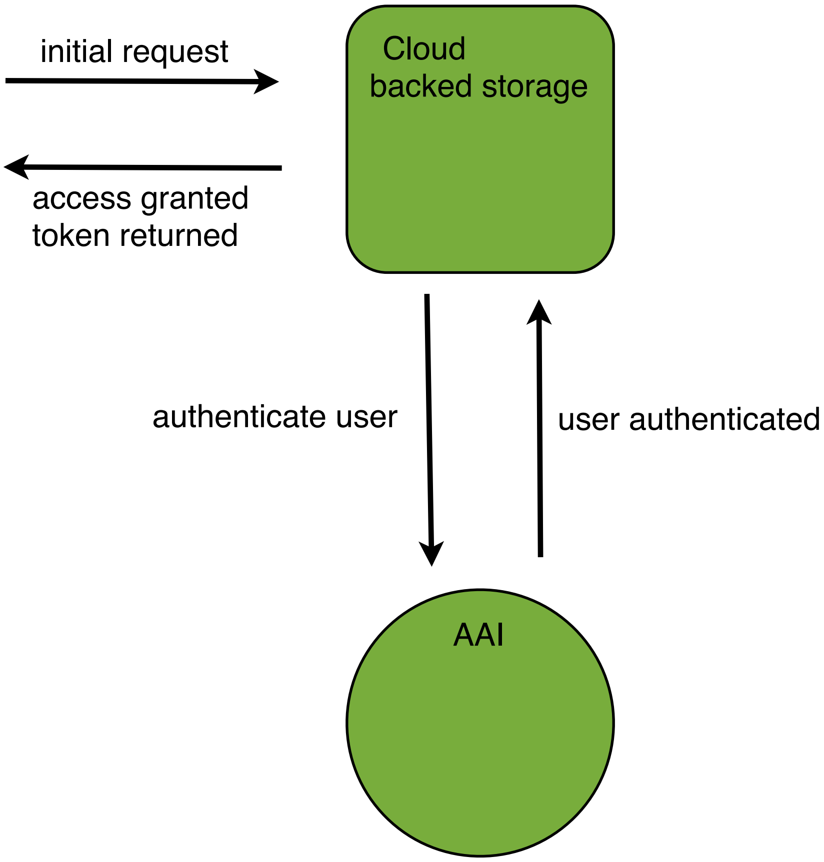
2. Architecture of cloud back storage

Taking all gaps and requirements into account we present an architecture based on open source components, open standards and some integration work (open sourced and based on open source) that will meet the challenges posted by the requirements and the current cloud backed storage infrastructure providers. In doing so the architecture also gives a level of independence of using one cloud infrastructure provider exclusively.

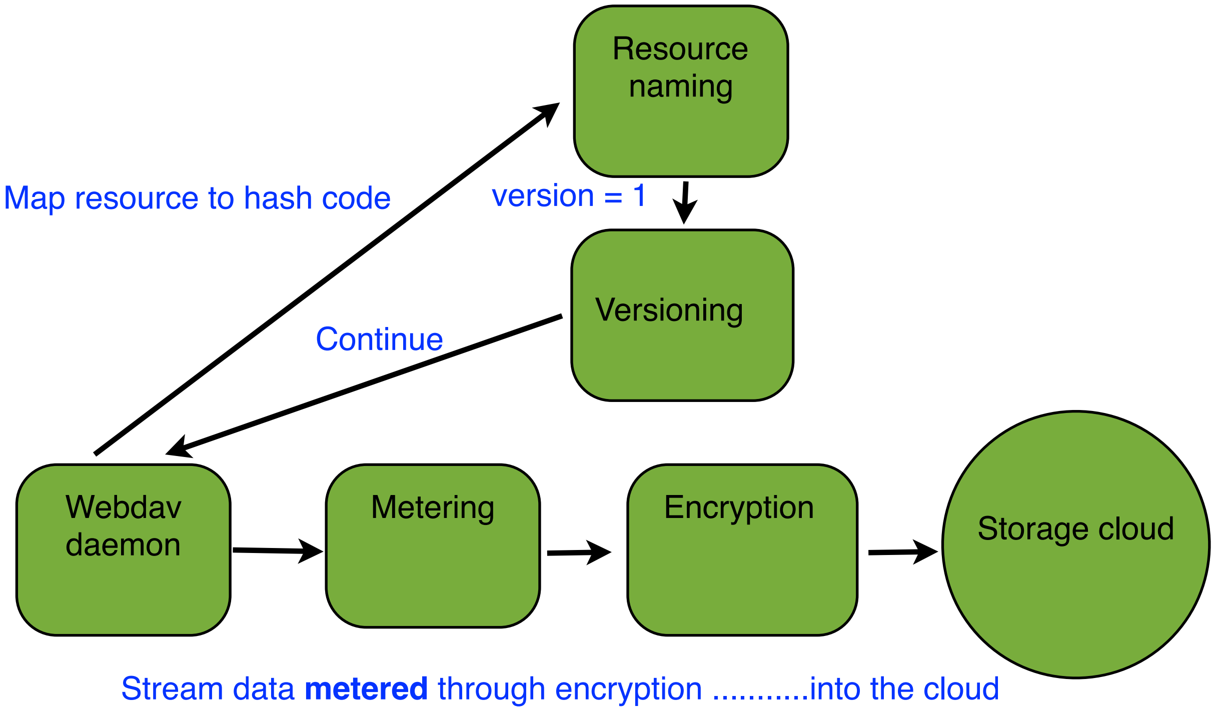
An overview

The below image gives an overview of the components involved in the cloud back storage service. Note that the components are assumed to run in an elastic computing environment (computing cloud). We will walk through the components based on some user scenarios. Then we will discuss some of the components and standards in a bit more detail.

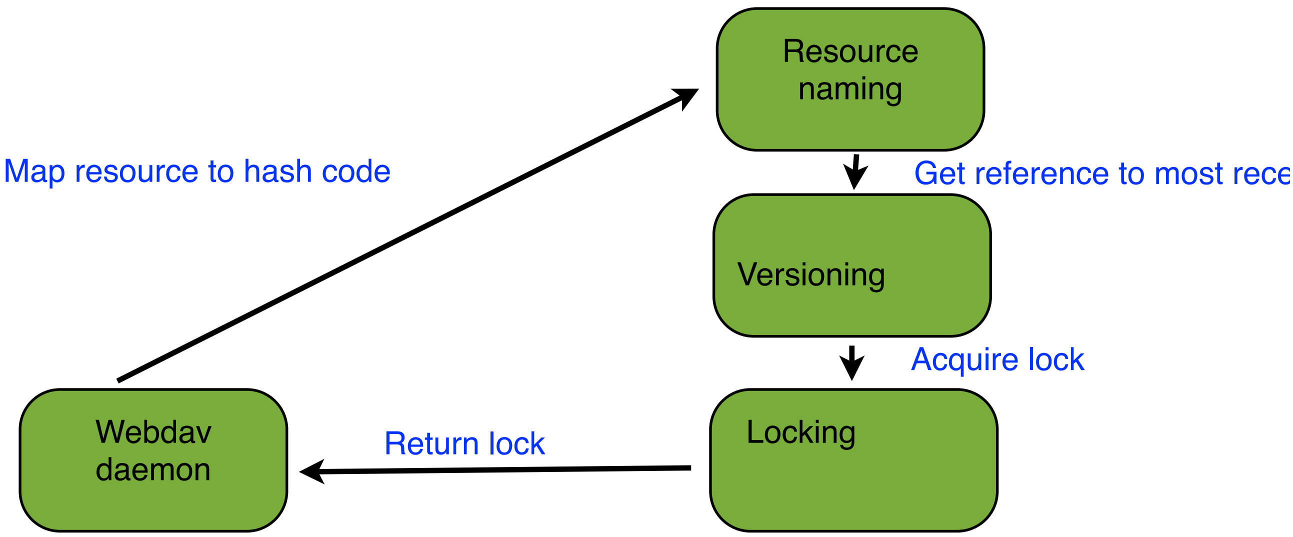
First, a user will need to enroll or activate its storage space. This is done via a simple application "protected" by the AAI. On successful authentication this application will give out a password (consider it a software token) that will allow the user to "mount" the network drive.



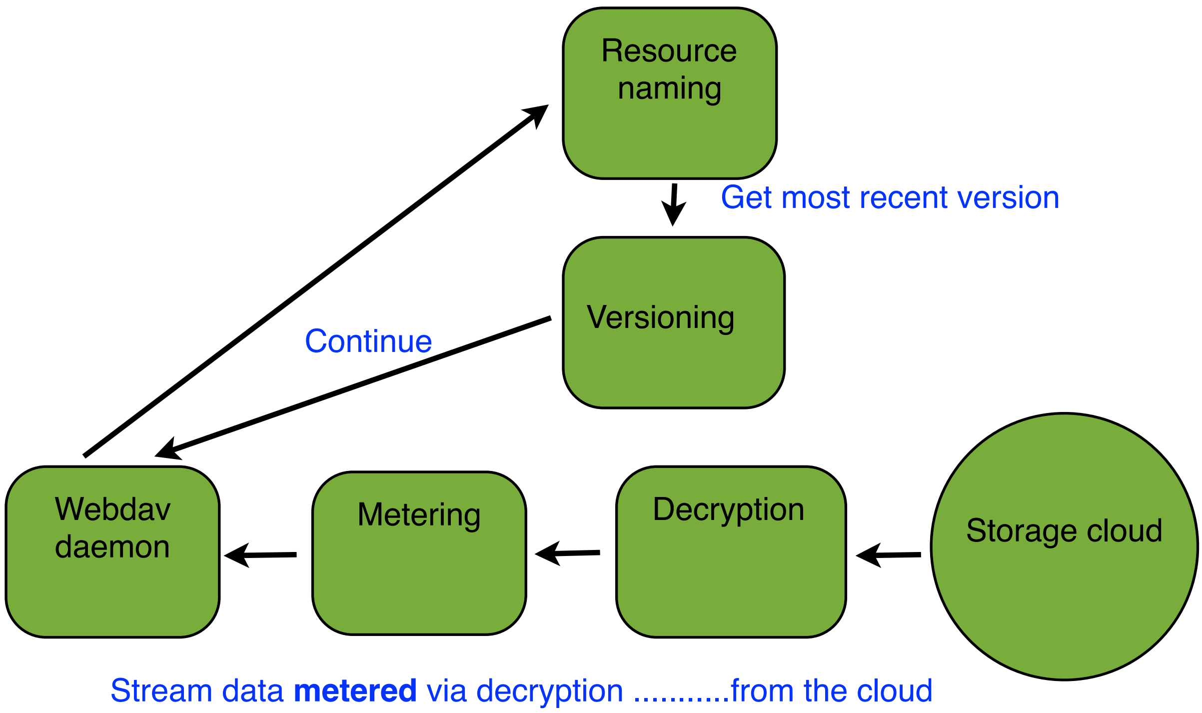
Next, the user mounts the network drive using standard operating system facilities. The most widespread HTTP based file system driver is WebDAV, eliminating the need for a special client. The user may create a file, which causes a version (1, as this is a new file) to be created in the versioning service. On creation of the file, the data is on the fly encrypted and metered.



The user can then share or lock the file via the respective services.



Downloading or opening a file decrypts it (again on the fly) and meters the outgoing transfer for the user that opens it.



3. Results

The above described architecture has been successfully implemented and has been in use on a small scale since Q4-2010. Approximate time spent was 5 man-months. The main lesson learnt are:

- true horizontal scalability (thus providing maximum elasticity) requires a radical and complete rethinking of application design. Having said that, most applications don't need true elasticity.

- cost is not insignificant but cost/benefit ratio is generally good

- WebDAV and the likes are available on many platforms but require significant testing. This is not related to clouds, but it is related to "making it easy accessible". It does explain why most clouds are API based, shifting the burden in many cases to the providing (implementing) party.

- Whenever a so-called NoSQL datastore is used, care must be taken to measure performance requirements and upper limits of e.g. memory caches.

- Clouds require education/training - otherwise support will be too big an effort.

7.5 Appendix V - Cloud experiments Finland

Author: Jukka Kommeri, Helsinki Institute of Physics Technology Program

1. Introduction

Clouds are new field of study and might provide aeasy to use computing resources for different proof of concept systems. We have been using virtualization in our work for a long time and for a while we have been looking into different management tools for our virtualized machines. Clouds provide an easy to use interface to manage virtualized resources and their users. Also as the cloud technologies gain popularity, it becomes an important field of energy efficiency. Because energy efficiency studies of computing systems is one of our most important fields, understanding how they works is vital for making them more energy efficient.

2. Pilot

As part of the Northern European Cloud project we have set up proof of concept cloud installations. We have had at our disposal a bit older servers that do not have the virtualization support that modern server have. Regardlessly we have managed to test the two main open source cloud middleware projects Open Nebula and Eucalyptus.

As the older hardware does not support virtualization we have had resolve to using Xen's paravirtualization. Xen provides an efficient virtualization platform with low virtualization overhead. Its shortcoming is that it requires modifications to virtual machine's kernel and drivers and is hence difficult or impossible to apply to non Linux operating systems like Microsoft Windows.

We have set up our clouds using different versions of Rocks operating system. Rocks provides us a scalable way to set up bigger cloud environments and to manage large pool of virtualization hypervisors. Rocks is based on CentOS distribution and is one of the last Linux distributions to have a support for Xen virtualization.

In our test clusters we have successfully tested auto3dem software that uses openmpi. Due to our limited resources, its large testing is still undone, but its functioning has been tested.

3. Conclusion

For the end user the current versions of open source cloud projects already provide an easy to use interface for the management virtual machines. They provide a way too choose between different level of resources and give good control over virtualized resources. As such they already provide non-business organizations with a decent system to easily set up test environments or different proof of concept systems.

Clouds are based on virtualization. Even though there is support for hardware that does not have the support for virtualization, the projects focuses on the virtualization technologies that requires it. Also current Linux distributions are focusing on the same technologies. Meaning that Xen's paravirtualization support is slowly dropping from all distributions. Out of the main distributions only CentOS seem to support it at the time of writing.

7.6 Appendix VI - A pilot virtualized batch facility - Denmark

Author: grid.dk

1. Pilot case

Computational high energy physics (HEP) requires very large, non-standard software installations. This last requirement makes it difficult to use all potentially available resources.

Here, virtual machine technology can be of help: Instead of distributing large software packages in binary format for various platforms or as source code, leaving it for expert to compile and install, simply distribute preconfigured virtual machine images.

2. Result

This idea has been implemented on a 24-core farm at CERN: A small library of virtual machine images was put in place and used for high energy physics applications and medical image analysis. For the case of HEP, the images were provided by the CernVM project, for medical image analysis, a minimal Debian image was used.

A combined batch and virtualization system was written from scratch. The batch part of the system was heavily tested by HEP users who ran 20'000+ jobs on this small farm.

The reason for writing a system from scratch is that we did not see any clear way of combining private cloud technology with old-school batch systems like Torque or SGE.

7.7 Appendix VII - Notur cost estimates - Norway

November comparisons between Amazon EC2 and Do-It-Yourself, from Notur, Norway.

PRICE COMPARISON BETWEEN AMAZON CLUSTER INSTANCES AND DO IT YOURSELF

1.00	Euro	=	8.00 NOK
1.00	USD	=	6.33 NOK

Since cloud is largely pay-as-you-go (but one can pay a fixed price upfront to get discount), we consider different usage (loads) of the instances/nodes

Amazon pricing taken from http://aws.amazon.com/ec2/pricing/

AMAZON CLUSTER INSTANCE (Quadruple Extra Large)	Fixed USD	Per hr USD	Per yr USD	Per 3 yrs USD	**USAGE** **100%** Price per yr / instance NOK	**50%**	**30%**	**0%**
Price per hour - pay as you go - USD		1.60	14016	42048	88721	44361	26616	0
Reserved instance (1 yr term) - USD	4290	0.56	9196	27587	58208	42682	36471	27156
Reserved instance (3 yr term) - USD	6590	0.56		21307	44957	29431	23221	13905
Spot Cluster Instance - USD (estim. 33% og default price/hr)		0.53	4672	14016	29574	14787	8872	0

NOTE:
Network costs: not included
User support: not included
Storage solution: not included
Spot Cluster Instance is not available (yet), but its price is estimated from spot pricing for other instance types.

DO IT YOURSELF

Idea: we buy and operate a cluster with nodes that are equal/similar to the Amazon Instances

Cluster Instance (Quadruple Extra Large): 1000 nodes

				USAGE **100%** Price per yr / node NOK	**50%**	**30%**	**0%**
Cluster Instance - node price (USD)	3000	USD	(1)	6330	6330	6330	6330
Electricity (NOK) Price kWh:	0.5	PUE: 1.75	(2)	3066	2683	2300	1916
Infrastructure: price per node			(3)	1333	1333	1333	1333
Personnel: operations cluster (850 KNOK/FTE)	0.5	FTE		425	425	425	425
Personnel: operations infrastructure (850 KNOK/FTE)	0.2	FTE		170	170	170	170
Sum Cluster Instances				11324	10941	10558	10175

NOTE:
Network costs: UNINETT (hidden)
User support and management: not included
Storage solution: not included

(1): This cost includes node price (2200 USD) + interconnect price Infiniband (700 USD) + other expenses (100 USD); Life time of cluster: 3-years

(2): Electricity usage per node. Assumption: 250 W idle (0% use) - 400 W max (100% use); PUE = Power Usage Efficiency (ideal =1,0)
(3): Example machine room: 100 sqm, building cost 20 MNOK, lifetime 15 yrs, 64 nodes/rack; 1 rack = 0,6x1,8 = 1,0 sqm (incl. service area)

This includes floor space, UPS, cooling infrastructure, cabling, external network, ...
Assumption: whole machine room is dedicated to this single cluster.
Then cost per node is approximated as building cost / lifetime / # nodes

OTHER CONSIDERATIONS

Amazon Cluster Instances are based on a single node configuration at the moment.
In case one restricts usage to jobs with upto 8 cores (i.e., jobs fit within single nodes/instances), then the node price for DIY drops with ca. 25% (no Infiniband), but the Amazon pricing drops more as one can use instances that are considerably cheaper

Linux instances, November 2010				USAGE			
				100%	50%	30%	0%
	Fixed	Per hr	Per yr	Price per yr / instance			
Micro – 1 CPU, burst 2 CU, 768MB RAM	USD	USD	USD	NOK			
Price per hour - pay as you go - USD		0.020	175	1109	555	333	0
Reserved instance (1 yr term) - USD	54	0.007	115	730	536	458	342
Reserved instance (3 yr term) - USD	82	0.007		561	367	289	173
Spot Instance - USD		0.007	61	388	194	116	0

Linux instances, November 2010				USAGE			
				100%	50%	30%	0%
	Fixed	Per hr	Per yr	Price per yr / instance			
Large - 2 cores	USD	USD	USD	NOK			
Price per hour - pay as you go - USD		0.034	298	1885	943	566	0
Reserved instance (1 yr term) - USD	910	0.120	1961	12414	9087	7757	5760
Reserved instance (3 yr term) - USD	1400	0.120		9608	6281	4950	2954
Spot Instance - USD		0.121	1060	6710	3355	2013	0

Linux instances, November 2010				USAGE			
				100%	50%	30%	0%
	Fixed	Per hr	Per yr	Price per yr / instance			
Extra Large - 4 cores	USD	USD	USD	NOK			
Price per hour - pay as you go - USD		0.680	5957	37707	18853	11312	0
Reserved instance (1 yr term) - USD	1820	0.240	3922	24829	18175	15513	11521
Reserved instance (3 yr term) - USD	2800	0.240		19216	12562	9900	5908
Spot Instance - USD		0.237	2076	13142	6571	3943	0

Linux instances, November 2010				USAGE			
				100%	50%	30%	0%
	Fixed	Per hr	Per yr	Price per yr / instance			
High-Memory Extra Large - 2 cores	USD	USD	USD	NOK			
Price per hour - pay as you go - USD		0.500	4380	27725	13863	8318	0
Reserved instance (1 yr term) - USD	1325	0.170	2814	17814	13101	11215	8387
Reserved instance (3 yr term) - USD	2000	0.170		13647	8933	7048	4220
Spot Instance - USD		0.162	1419	8983	4492	2695	0

Linux instances, November 2010				USAGE			
				100%	50%	30%	0%
	Fixed	Per hr	Per yr	Price per yr / instance			
High-Memory Double Extra Large - 4 cores	USD	USD	USD	NOK			
Price per hour - pay as you go - USD		1.000	8760	55451	27725	16635	0
Reserved instance (1 yr term) - USD	2650	0.340	5628	35628	26201	22430	16775
Reserved instance (3 yr term) - USD	4000	0.340		27293	17867	14096	8440
Spot Instance - USD		0.415	3635	23012	11506	6904	0

Linux instances, November 2010				USAGE			
				100%	50%	30%	0%
	Fixed	Per hr	Per yr	Price per yr / instance			
High-Memory Quadruple Extra Large - 8 cores	USD	USD	USD	NOK			
Price per hour - pay as you go - USD		2.000	17520	110902	55451	33270	0
Reserved instance (1 yr term) - USD	5300	0.680	11257	71256	52402	44861	33549
Reserved instance (3 yr term) - USD	8000	0.680		54587	35733	28192	16880
Spot Instance - USD		0.830	7271	46024	23012	13807	0

Linux instances, November 2010

High-CPU Medium - 2 cores	Fixed USD	Per hr USD	Per yr USD	USAGE 100%	50%	30%	0%
				Price per yr / instance NOK			
Price per hour - pay as you go - USD		0.170	1489	9427	4713	2828	0
Reserved instance (1 yr term) - USD	455	0.060	981	6207	4544	3878	2880
Reserved instance (3 yr term) - USD	700	0.060		4804	3141	2475	1477
Spot Instance - USD		0.057	499	3161	1580	948	0

Linux instances, November 2010

High-CPU Extra Large - 8 cores	Fixed USD	Per hr USD	Per yr USD	USAGE 100%	50%	30%	0%
				Price per yr / instance NOK			
Price per hour - pay as you go - USD		0.680	5957	37707	18853	11312	0
Reserved instance (1 yr term) - USD	1820	0.240	3922	24829	18175	15513	11521
Reserved instance (3 yr term) - USD	2800	0.240		19216	12562	9900	5908
Spot Instance - USD		0.246	2155	13641	6820	4092	0

Linux instances, November 2010

Cluster GPU instances	Fixed USD	Per hr USD	Per yr USD	USAGE 100%	50%	30%	0%
				Price per yr / instance NOK			
Price per hour - pay as you go - USD		2.100	18396	116447	58223	34934	0
Reserved instance (1 yr term) - USD	5630	0.740	12112	76671	56155	47948	35638
Reserved instance (3 yr term) - USD	8650	0.740		59285	38768	30562	18252
Spot Instance – USD: N/A		0.000		0	0	0	0