HEAD IN THE CLOUDS

Hans-Christian Boos

5. Dezember 2008

Much has been written about clouds and cloud computing. Too much to get a good understanding of the possibilities and opportunities offered by this new approach to maximizing hardware use and minimizing programming effort. Therefore this article is designed to give a brief overview of the basic idea behind cloud computing, the business use and the steps that seem natural in order to make cloud computing an economic factor rather than a pretty buzz word in IT marketing. The first step I call the resource cloud shows the reuse of hardware through virtualization. This approach is already generally available. As the second step the article covers the possibilities of parallel computing through a new programming style enabled by the cloud idea – resulting in establishing the real cloud as I choose to call it. To wrap up this brief overview on a topic that may change much of our current view of IT, the article shortly examines what can and cannot be done to migrate current IT infrastructures and architectures towards a cloud centered computing network.

THE TWO KINDS OF CLOUDS

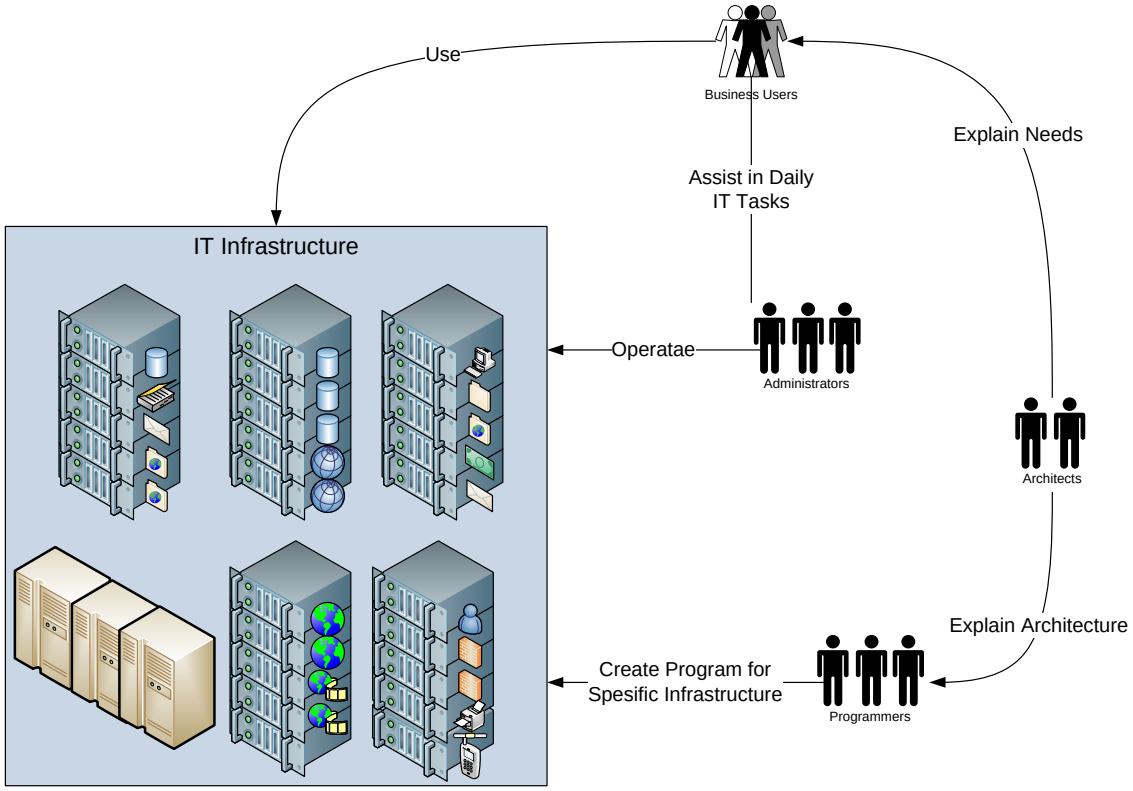
As stated briefly there are two main stream interpretations of a cloud. They actually build on one another. I will call them "resource cloud" and "real cloud" in order not to get involved into any religious discussions on buzz words. Even though the resource cloud is the logical first step in cloud computing the concept of the real cloud was talked about first. For logic´s sake, let us start by getting into the functions of a resource cloud.

FROM SCROOGE MCDUCK TO BANKING AND FROM DATACENTERS TO THE RESOURCE CLOUD

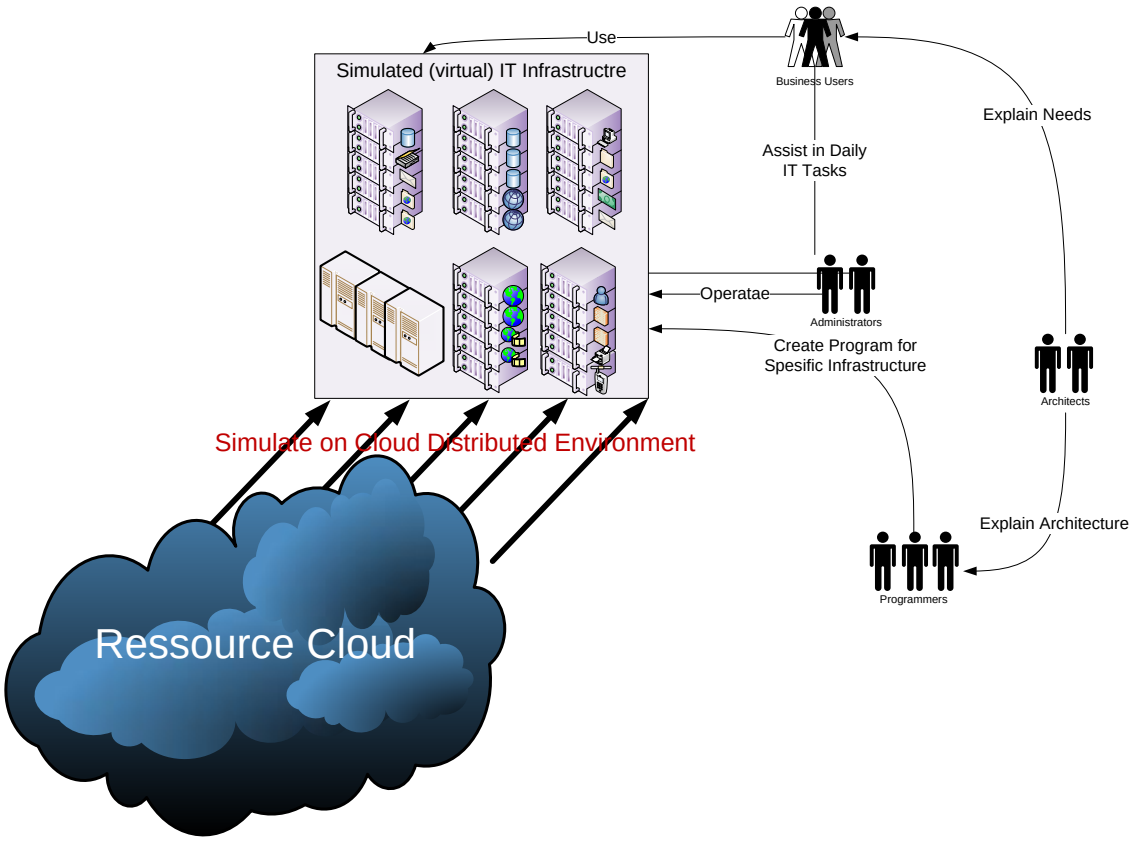
Imagine you do not have to care about server resources anymore. Computation power (CPU) and memory (RAM or storage) are available to you on demand. These resources would come to you just by requesting them and would disappear again after you were done with them. Just like electricity that flows when you plug your laptop into the socket and more electricity flows, when you plug in your hairdryer as well – then when you turn those things off – or unplug them – the electricity stops flowing. That is the basic idea of a resource cloud.

If we stay with the electricity example, you just use the IT resources and no longer have to buy the "power plants" (i.e. servers, storage networks or whatever else there is) that produce the resource. You don´t even have to know where these power plants are. You simply plug something into the socket and the resource starts coming to you at just the rate you need it and just the time you need it for.

Not knowing where your data is processed or stored seems to be a major issue for some people. I have been talking to grown managers of multibillion Dollar companies who virtually wanted to "sit on their IT". This feels a little bit like Mr. McDuck who does not trust anyone and keeps his gold in his own storage (not very effective, if you remember correctly). Then again despite the financial crisis still most people have no problem with banks handling their money storage, their transactions and whatever else is to be done around today´s global fluid of life. So moving from "having your IT safely tucked in the basement of one or many of your buildings" to using a resource cloud is very similar to moving from McDuck style money management to modern banking – mentally challenging but definitely a good idea!.



The idea of resource clouds are put out on the market in many different forms. They may sometimes come with proprietary interface. More often they appear as virtualized “machines” that you can use, grow and shrink on your request. Some of these virtual machines even offer additional features (or spoken abstractly: value added) to differentiate themselves from other players on the market. With this kind of model just using a cloud is about as easy for IT people as plugging the hairdryer in is for everybody else. However one must carefully study the service and service quality a supplier of such a resource cloud is offering: Some suppliers (like amazon.com) offer clouds to cheaply sell off their excessive computing power. IT resources these suppliers have to keep in stock for handling peak usage periods. At Amazon for example such a peak usage period is Christmas time. So when your business also has a great deal of IT action – e.g. transactions – going on at this time, Amazon is probably not the right cloud supplier for you, because they will serve their own business before they serve yours. And yes, they do not rip you off, but tell that to you right when you sign up. Such a resource cloud is a great idea: economically, because you only pay for whatever resources you use and environmentally because this might actually increase the level of actual IT usage (today under 10%) up to a level of maybe 50%.



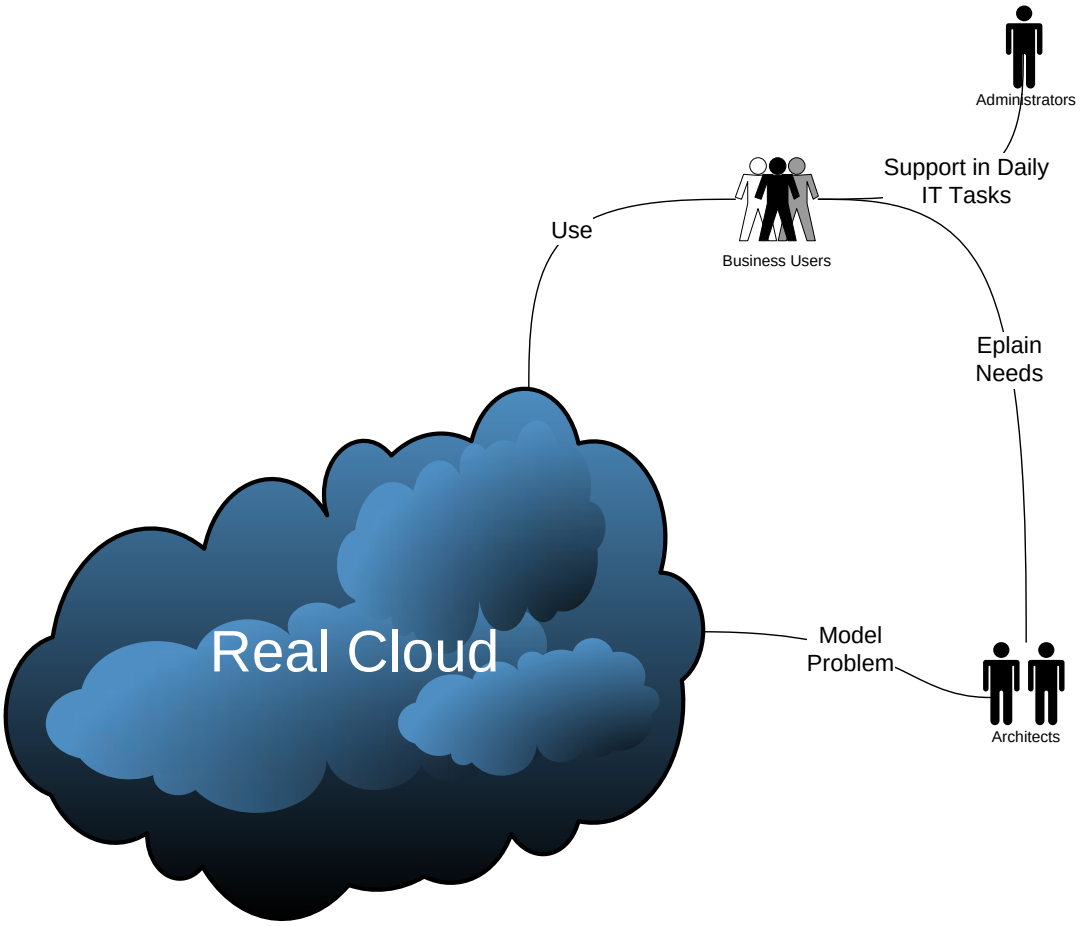
Using a resource cloud might be a mental challenge for old style management, but if you can find a supplier offering the kind and level of service you need, it is surely what you should look into.

The Master Control Program or What the Real Cloud is all About

After having looked into what everybody can use today, we move on to take a look into computational future. The idea of a cloud is to make IT-power become available on demand, and a resource cloud does a good job of that in today´s terms of IT. Still a resource cloud is just the simulation of "today´s" IT landscape onto a somewhat distributed model. The limitation of virtualization – used to produce the resource cloud – is that one machine being simulated also resides on one machine. The same machine maybe host to many such simulations, but if the simulation takes up all the resources of this machine, all other hosted simulations must be relocated. And if the resource demands of the virtual machine we are using outgrows its host, then our virtual machine has to be relocated to a more powerful host in the current model of a resource cloud. A next step could be to advance virtualization in order to spread a simulation over several resources. This may pose some technical challenges as an internal bus of a machine is much faster than anything that could connect two resources in different locations today, but I am sure someone will come up with a faster way to connect things. But there is a physical limit to connection speed and it is most likely that data transfer within one physical unit will be faster than transfer between such units for the foreseeable future.

So if this is not the alternative to salvage the computational power available in the hardware around the world, what could be? Well, an approach would be to reduce the size of the chunks of IT power that are requested. What would happen if we could just request single instructions to be performed on and not whole environments to be provided by a specific host? Anyone who has dealt with the development of parallel algorithms will now have a big smile on the face. The idea of making computational power available by instruction is great. But who can write a program that actually does such distributed instruction requests? Nobody really can. If we examine the challenges of today´s software projects we will find parallel processing at the top of the list. Even multi threaded

environments – like most modern operating systems are – are above the capacity of the majority of developers. And if someone really understands how to do “distributed computing” it is very likely that we are not going to find any knowledge or even interest in the business process to be supported in the same person.



So if we want to get any further in producing a really distributed environment we will be looking at IT support in order to make it work. Just like schedulers in today´s operating systems give CPU power to all the processes and applications running on one machine, a cloud management software will have to distribute instructions throughout a network of computers. This may sound much easier than it actually is. A scheduler – and a large, really cool and economically important project like the Linux operating system was started by developing a new and more efficient scheduler – is a program that lets all processes running in one environment execute their commands on the available hardware sequentially. This means the scheduler swaps a program into execution, lets it execute for a certain time and then exchanges the program by the next one in the queue. The scheduler has to be very effective in swapping in and out as well as prioritizing the programs to be executed. A good scheduler does this with as little overhead as possible thereby reducing the amount of time (the size of the timeslice) each program must run. In the end this gives the user the impression that all programs attached to one scheduler are executed simultaneously.

The main difference between a scheduler and the kind of “master control program” needed to distribute execution of single instructions onto a cloud network is, that a scheduler does not have to “understand” the programs whose execution it controls. In order to distribute the execution of a program into parallel execution the “master control program” has to know what instructions may be executed in the parallel and which instructions or blocks of instructions have to be executed sequentially – without a programmer explicitly identifying these “critical sections” or predicting “race conditions”.

This may sound like a great challenge to computer science, but there are actually distribution algorithms already out there that can do the job. These have to be refined and improved, but the basic problem of really distributed computing on a low level has been solved. This was the foundation for the hype on "Grid Computing" or "the Grid" you may have read about a couple of years ago. The problem with these approaches is, that the programmers have to explain how to execute their programs in a parallel environment. To have such an algorithm actually decide by itself what can be submitted for parallel execution the level of programming has to go up to the level of an abstract model. This means where we actually describe what is to be executed in today´s programs a "real cloud program" will rather describe the results of the execution and the way things are calculated in a symbolic approach. This has been postulated by the geeks and gurus of software engineering for a long time, but is unfortunately not applied to most programming projects.

Implementing the real cloud – a mechanism that would most likely push the usage of available hardware to a level of 80% or higher – means writing or rather modeling software for a parallel environment. Thus legacy software would have to be migrated to a cloud environment. The latter being unlikely there will be a co existence of real cloud and resource cloud applications in a medium perspective.

Lots of problems in real cloud management are yet to be solved. Some technical, like how can such a master control program become a distributed application itself and how it is to deal with a dynamically changing network of available resources - some of them psychological or organizational.

TRON – A VISION OF REAL CLOUD COMPUTING

Obviously there is quite a bit of room for improvement between what is available today, what is talked about in the press and what the actual concept of cloud computing can take us to. Philosophically speaking real clouds will take us into the brave new world of computing, where we just come up with the ideas and some magical entity makes it come true. The resource cloud everybody is working on is a good step on the way, but it is not even half way yet. Economically speaking a resource cloud will enhance the usage grade of IT from 10% to maybe 50% and the real cloud will actually bring it up to 80%. So when we are done implementing clouds we will get eight times the computation power at the same investment level, respectively saving the money.

All this technical development was foreseen by the motion picture director Steven Lisberger when he produced the movie TRON in 1982. In this film there is a huge network of computers forming a virtual world where programs lived and worked. In this world program execution and resource allocation was centralized. The world was dominated by the "MCP" – the master control program. In this film the MCP was the "bad guy" as it was a hindrance to free programs. Well we do have free programs today, and we are moving into the direction of real clouds with some sort of a master control program managing the whole scenario.

THE LEGACY OF NON-CLOUDS SOFTWARE

So, what happens to IT after we have finally solved all the problems of managing a real cloud? Surely all the predictions about IT becoming a commodity and about the few suppliers who will actually sell cloud resources to us are true, but all that does not answer the question of what really happens to the IT that is in place then.

The answer is just as frustrating as simple. All the IT in place then will have to be replaced!

It will have to be replaced, because it is actually written as computer code or code close to computers. A cloud controller – as briefly explained above – does not need code but a software model (the idea behind the program). So this means that in order to use all the advantages of a real cloud, all the software out there will have to be rewritten or at least it will have to be reengineered. Does this sound familiar? Does the word of "the age of hosts is over" come to mind to any of you? Well as the IT in

place then will work well and as it probably will not have become much better documented or modeled it is very unlikely that clouds will replace today´s understanding and management of IT right away. Clouds will be relevant to new application and new kinds of application design. The economic question to be answered is whether there are enough programs rewritten or redesigned to finance the tremendous investment required to build up clouds. Or will the first wave of clouds go bust with a billion dollar loss and only the second wave will then finally bring effectiveness in IT usage to all of us?