Energy Optimisation

Barry Denby

Griffith College Dublin

April 22, 2021

What will be discussed

- In this lecture we will discuss the issue of Energy Optimisation in a cloud.
- A cloud as it comprises large amounts of computation power will require power and cooling on a large scale.
- However this will conflict with performance optimisation goals.
- This lecture will explore the interplay between both and how research in this area is trying to solve this problem.

Why Energy optimisation is important to a cloud

- In large datacentres there are significant requirements in terms of power and cooling to keep the datacentre running.
- As such a significant portion of a budget for a datacentre is devoted to the ongoing power and cooling requirements of the datacentre.
- If it is possible to minimise the power and cooling requirements for such a datacentre then one of two things can happen.
- The cost can be passed onto the consumer by reducing prices or by keeping pricing the same profit can be increased.

Why Energy optimisation is important to a cloud

- There are multiple ways in which power and cooling can be kept to a minimum.
- First and foremost is picking a geographical location that is generally cool all year round.
  - Ireland generally is hence why Amazon, and Google have datacentres here.
  - Facebook has a datacentre in Sweden 110Km south of the Arctic Circle.

Why Energy optimisation is important to a cloud

- If possible the geographical location should have access to cheap electricity too e.g. hydroelectric / nuclear power.
  - Amazon has datacentres in Oregon and West Virgina which have the 11th and 13th cheapest electricity prices in the US by state (Data is from 2011).

Why Energy Optimisation is important to a cloud

- The geographical decisions can only be made everytime a datacentre is commissioned and built.
- However there are ways and means power and cooling consumption can be reduced after a datacentre has been built.
- Some of these involve the physical hardware.
- However a lot of the research focus is on the cloud management of virtual machines.
- We will explore both but the latter in more detail.

Physical hardware

- In terms of physical hardware to achieve a reduction in power will require the use of installing more powerful physical nodes in place of less powerful ones.
- Or to change physical nodes for more energy efficient versions.
- At the moment with Moore's law still going this can be done on a roughly 18 to 24 month basis as computation power is still increasing year on year.
- There is also currently interest in using more energy efficient architectures like ARM in these environment as they require a lot less power than x86 does.

Physical hardware

- Finally the last thing that can be done is to keep as much hardware as possible in a low power or off state.
- This way it will consume the least amount of power but can be woken up as and when it is needed.
- This is where the OS managing the cloud comes into play as it will decide when nodes can be put to sleep or when they are to be woken up.
- Will be the main interest for the rest of the lecture.

Switching off hardware

- The reason why deciding when to turn on or turn off physical pieces of hardware is of such interest in research is that the strategy for managing this will have a large influence on the power and cooling required for the datacentre.
- If we can keep the minimum amount of hardware powered on at all times we will minimise power and cooling requirements.
- However, this is difficult to maintain in a cloud as the workload in the cloud fluctuates every second of every day.
- To discuss this more precisely we will define a few terms.

Physical hosts and Virtual Machines

- A Virtual Machine represents a container with an operating system, libraries, and applications that can run on a Virtual Machine Monitor.
- A Physical Host refers to any physical piece of hardware that is capable of hosting and running a Virtual Machine through the relevant Virtual Machine Monitor.
- A Virtual machine may only reside on a single Physical Host.
- However a Physical Host may host many Virtual Machines.

The energy optimisation problem

- Thus the energy optimisation problem is one in which a cloud must adaptively allocate virtual machines to physical hosts in the context of unpredictable workloads.
  - paraphrased from the paper "Optimizing virtual machine placement for energy and SLA in clouds using utility functions".
  - we will refer to this paper again throughout the lecture.
- Such that we use the minimum number of physical hosts and therefore the minimum amount of power and cooling.
- However there are conflicting objectives here between Energy Optimisation and Load Balancing.

Load balancing

- Load balancing when applied to a cloud seeks to minimise the response time of every running application.
- It does this by implicitly (through work queues) or explicitly (by using a direct load balancer) allocating work to the host that currently has the lowest overall load.
- For example if we have a set of four hosts with an overall load of 2.0 then load balancing ideally would aim for all hosts to have a load of 0.5.
- Where all hosts have 50% spare capacity in the event of a spike in demand.

Load Balancing

- The advantage of such an approach is that response times are minimised as lightly loaded hosts will respond quickly.
- Also the spare capacity enables the approach to handle temporary spikes relatively well provided there is enough spare capacity.
- The disadvantage of the approach is that there is spare capacity that is consuming power and cooling for no tangible benefit.
- i.e. all the time there is spare capacity the cloud is losing money by using power and cooling to keep this unused capacity running.

Energy Optimisation: The ideal

- Energy optimisation in the ideal case would seek to keep as many hosts as possible at a load of 1.0 with all other hosts having a load of 0.0.
- The hosts with a load of 0.0 will be put in standby or powered off to minimise their power and cooling requirements.
- It will only power on a host if there is additional load and there is no host with spare capacity.
- See the example on the following slide.

Energy Optimisation: The ideal

- Taking the previous case if we have 4 hosts with a total load of 2.0 energy optimisation would seek to have two hosts with a load of 1.0 each and the other two with a load of 0.0.
- It would then suspend or power off the two hosts with the load of 0.0.
- Should additional load be added and the two active hosts still have a load of 1.0 it will then wake up one of the suspended or powered off hosts to accept this load.
- It will keep adding new loads to this host until all nodes have a load of 1.0 before waking up another host.

Energy Optimisation: The ideal

- Consequently when the overall load reduces to 1.0 less than the total number of active hosts the approach will try to move around all workload such that it can suspend or power down another node.
- The advantage of such an approach is that power and cooling requirements are kept to a minimum at all times.
- The issue however is that for this minimisation response time is sacrificed.
- Care must be taken here as SLAs must be upheld at all times.

Energy Optimisation vs Load Balancing

- Thus energy optimisation and load balancing are conflicting goals.
- One seeks to have all physical hosts available at all times.
- The other seeks to have the bare minimum of hosts active at all times.
- Thus a balance between the two approaches must be achieved to keep the cloud running in a reasonably responsive manner and able to react to load spikes.

Energy Optimisation vs Load Balancing

- However care must be taken with such approaches as there are two potential pitfalls that such approaches can run into.
- Workload fluctuations that cause physical hosts to switch between active and inactive states quickly and frequently.
- Virtual machines being shifted around from host to host quickly and not given much time to run on each host.
- We will explore this in more detail by referencing the research paper cited earlier.

The Energy Optimisation problem

- The energy optimisation problem can be summarised as follows
  - Given N physical hosts.
  - M virtual machines and associated workloads.
  - On each iteration.
    - Construct a Virtual Machine to Physical Host mapping that satisfies the following conditions.
    - Minimises energy usage (and by consequence cooling).
    - Minimises SLA violations.
    - Minimises VM transfers from one physical host to the next.

The Energy Optimisation problem

- However as the authors rightly point out a cloud will have resource constraints.
- Thus there is an additional step in which the VM to PH placement will be evaluated against the resource constraints.
- If it breaks the resource constraints then it is rejected and a new placement must to generated.
- An example of a utility function that the authors use to decide on a VM to PH placement is shown on the next slide

Utility Function used

- $Utility(a, t) = Income(a, t) - (EstimatedEnergyCost(a, t) + EstimatedViolationCost(a, t) + PDMCost(a, t))$
  - a represents a VM to PH mapping.
  - t represents time.
- What this function aims to do is by minimising the total cost of Energy, SLA violations, and the performance loss from migrating VMs from one PH to another, cloud profit is maximised.
- We will discuss the terms in more detail on the next slide.

Utility Function used

- $Income(a, t)$ represents the total VMs that are being hosted on the cloud and the income that is generated from that set of VMs.
  - Time is included here as the VMs will increase and decrease over time.
- $EstimatedEnergyCost(a, t)$ is a function that will approximate how much energy will need to be used in order to keep the PHs running that support this VM mapping.
  - Again time is included here as mappings will change over time.
  - Also as the cloud will be somewhat heterogeneous in nature some PHs may be more energy efficient than others.

Utility Function used

- $EstimatedViolationCost(a, t)$ is a function that will approximate how many SLA violations will occur in this mapping, along with the cost associated with those violations.
    - Most cloud provider SLAs will include penalties imposed on the provider if certain performance metrics are violated over time.
    - Usually this will result in the provider refunding some of the clients cost directly to them.
    - Ideally SLA violations should be 0 but this is not possible in a large cloud.

Utility Function used

- $PDMCost(a, t)$ is a function that will approximate how much performance degradation will occur when virtual machines migrate from one PH to another.
  - It takes time for a VM to be suspended and transferred to another PH before being made active again.
  - Again in an ideal world VMs would not need to move thus there would be no performance loss.

Utility Function used

- Of course these functions have a fair amount of complexity and a lot of code and we will not go into a lot of detail about them here.
- The $Income(a, t)$ function is relatively easy to compute. If you know what kind of virtual machines there are and also the cost associated with those virtual machines an overall income can be easily computed.

Utility Function used

- For the $EstimatedEnergyCost(a, t)$ we first need to compute the expected CPU usage of each PH in our assignment and then using a CPU to power model we then compute the amount of energy used by each PH.
- In general PHs with 100% cpu usage will be favoured.

Utility Function used

- For example in the paper using the SPECpower benchmark on two servers the following data was obtained about power consumption.
  - HP ProLiant G4 0% CPU $\rightarrow$ 86 watts, 100% CPU $\rightarrow$ 117 watts (36% increase in power).
  - HP ProLiant G5 0% CPU $\rightarrow$ 93.7 watts, 100% CPU $\rightarrow$ 135 watts (44% increase in power).
- Thus idling and low loads are very inefficient and wasteful.
- In terms of $EstimatedViolationCost$ it is necessary to first determine the total demands of all VMs in the current assignment.
- Then if the total demand is greater than the current supply we need to determine which VMs are in violation.

Utility Function used

- Once the VMs in violation has been calculated and we know what kind of violations they are, a cost can be determined for each violation and an overall cost computed.
- For the *PDMCost* function it is necessary to compute which VMs will migrate in the current assignment.
- An estimation will need to be made to determine how long it will take for those VMs to migrate and the cost associated with that.
- Once a cost has been generated for all VM migrations they are all added together to produce a result.

How a placement is decided

- When all of these functions have been implemented the approach will then use a genetic algorithms approach to evaluate the best mapping.
- By using the principles of the theory of evolution and survival of the fittest.
- The process is a stochastic search and the best result will be used for the next assignment of VMs to PHs.
-

Results

- In simulation this has been found to reduce the number of SLA violations, VM migrations, PH shutdowns, and reduce energy usage.
- However it has yet to be tried in a real world environment.
- That said this will reduce energy and cooling usage further by reducing the amount of time spent activating and deactivating PHs.