Failure is an option

Andrew Rice Sherif Akoush Andy Hopper
Computer Laboratory, University of Cambridge
[firstname].[lastname]@cl.cam.ac.uk

Abstract

Reducing the level of redundancy in a datacentre's power and cooling infrastructure can have a big impact on reducing overall energy costs. One means to reduce this overhead is to expect failures and adapt to them rather than attempting to eliminate them at all costs. High-level specification of services within a datacentre combined with technologies such as migration might provide us with the flexibility to run closer to the wire and adapt to infrastructure failures if they occur.

We argue that Service Level Agreements (SLAs) currently act as a disincentive to exploiting this flexibility and we suggest a more customer-centric specification which gives service providers the freedom to provision adaptively and provides incentives for both parties to work towards an appropriate service level for the client's business needs.

Specifying these agreements in machine-readable form is an important challenge and would provide two benefits: reaching, and relying, on these agreements can be made easier by modelling of the expected emergent behaviour of both parties; and integrating the specification of these new agreements into service declarations will allow us to begin to develop tools for orchestrating optimal adaption when failures occur.

1. Introduction

Our computing infrastructure is composed of a wide range of devices and infrastructure. Computing hardware such as servers, network infrastructure, storage devices, client machines and terminals are supported by infrastructure systems such as Uninterruptible Power Supplies (UPSs) and cooling systems. The energy consumption in the US due to servers and associated cooling systems alone has been estimated at 1.2% of total demand (Koomey 2007). Furthermore, the manufacture of microchips is a particularly energy intensive process. If we assume a three year operational lifetime of continuous use then a third of all the energy used by a server is in the manufacture (Williams 2004). Computing is consuming an ever increasing amount of energy. However, we believe there is significant scope for improving efficiency. We see the construction of an optimal digital infrastructure as a key research challenge of the future (Hopper and Rice 2008).

Modern datacentres have experienced rapid growth in size and energy consumption and now represent a significant proportion of our computing platform. In Section 2 we show that there is particular inefficiency in these large datacentres due to the support infrastructures (such as cooling and power systems) which support the high-reliability computing services we have come to expect and rely upon. Conventionally we justify this inefficiency because any failure of this infrastructure is viewed as a disaster scenario. However, we believe a datacentre could instead adapt to failures if they occur and therefore run closer to the wire. This adaption is already possible to a large extent using technologies such as virtualisation (Section 3). However, current service-level agreements present a key obstacle. We highlight shortcomings of conventional availability guarantees and suggest an alternative way for phrasing service agreements (Section 4). Quantifying, specifying, and analysing the emergent behaviour of these agreements is a vital first step towards an adaptive datacentre.

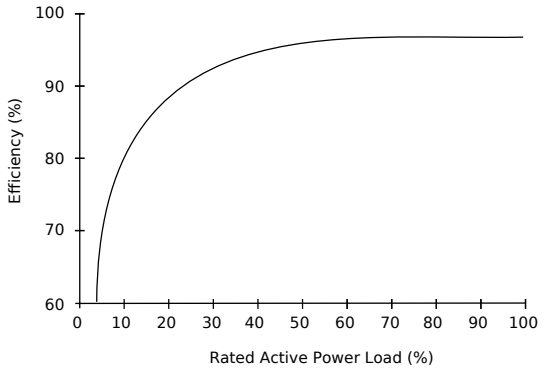

Figure 1. UPS Efficiency (high efficiency mode) (Ton and Fortenbury 2005).

2. Inefficient reliability

Datacentre reliability is often measured using the Tiering system (Turner et al.). A Tier 1 datacentre tolerates unavailability of the main electricity supply through the provision of an Uninterruptible Power Supply (UPS) and on-site generators. At the highest level, a Tier 4 datacentre provides redundancy at all points of the infrastructure through (independent) dual power supplies to every server on the machine floor. The two crucial design features of a Tier 4 facility are fault tolerance and maintainability. The facility can tolerate a failure of any single sub-system without any interruption in service and any single sub-system can be taken off-line for maintenance without interruption in service. Regularly servicing a datacentre's sub-systems provides a significant increase in reliability.

To highlight the inefficiency introduced, consider the efficiency curve of a UPS system (Figure 1). The vast majority of UPSs in use today are continually under active load and so the power distribution system is continually exposed to any inefficiency. We would ideally like to operate above 60% utilisation level due to the steep fall-off in efficiency at lower levels. This is problematic to achieve because simple over provisioning to cope with demand fluctuations or future growth can push us below this level. If one now adds a redundant live UPS for fault tolerance (a move from Tier 1 towards Tier 2) then we halve the utilisation—your UPS systems are in parallel and each must be capable of the full load if needed. If we move to a dual power system (Tier 4) then the utilisation of the UPSs halves again. The utilisation of each UPS

is now only 15% which in turn means 25% of our input power is wasted.

Achieving fault tolerance through added redundancy creates serious inefficiencies in our datacentres.

3. Adapting to faults

Rather than homogeneously provisioning capacity throughout the datacentre we could instead divide our systems up in to a number of independent units. The intention is that a fault in a single system should cause a reduction in capacity rather than a total failure. This gives the opportunity for adaption. Machines providing services with low availability guarantees might simply be switched off, diverting capacity to key services. Alternatively, services might be migrated to a smaller working set of machines perhaps compromising response time but maintaining availability.

Many key technologies with which to achieve this are already available. Recovery Oriented Computing (Patterson et al. 2002) considers how to minimise the recovery time after failure has occurred through techniques such as recursive restartability (Candia and Fox 2001) in which the system understands service dependencies. Virtualisation technologies such as Xen support the live migration of a running service from one physical machine to another (Clark et al. 2005). The migration approach can also be exploited to efficiently maintain a replica of a running virtual machine (Cully et al. 2008). This option provides redundancy at the logical level.

4. Smooth cost SLAs

Having highlighted efficiency issues and indicated the possibilities available for smooth adaption to failures we now discuss the impact this has on Service Level Agreements (SLAs). In light of their current growth and popularity we consider a web-services or Remote Procedure Call application in which a user makes an asynchronous request to a service which subsequently replies with a response.

We consider three principle actors:

- **The service provider** offers a computing platform for executing a service;
- **The users** make requests to the service and expects responses;
- **The client** contracts a service provider to provide a particular service for users.

The key goal of an SLA is to permit the client to specify the service level which should be provided to users. This is commonly an economic tradeoff in which the client chooses the right balance between the business benefit of a high service level and the high cost of sourcing it from a service provider.

As a simple example we consider the scenario in which a service provider is contracted by a client to host a website which will be selling tickets for an event. A conventional SLA might include:

- an availability guarantee;
- the maximum number of simultaneous users the site will support; and
- details of financial penalties and remedial actions for violation of the agreement.

This form of agreement would seem to suit the service provider because it makes provisioning straightforward but it does not suit the client. We imagine that the client might be happier specifying that:

- for each user who successfully requests the website the site the service provider gets paid some amount;
- for each user who requests the site and it fails to respond appropriately the service provider is fined some amount.

This model is closer to the cost analysis done by the business—what is each user worth to us? and what is the cost of not providing the service when they need it?

An agreement in this form has a number of benefits. Firstly, it aligns economic incentives with expected behaviour. For example, under the original agreement, if the site becomes overloaded the service provider simply incurs the penalty clause. In the second agreement the service provider is given an incentive to provide a good service to as many users as possible and no service to the remainder rather than almost no service to all users. Secondly, it allows the service provider to provision more efficiently. The cost of providing a certain level of fault tolerance in the datacentre can be compared with the cost of a particular adaption strategy when failure occurs.

Assigning a single price to a serviced request and a single fine for an unserviced request might be insufficient to express the client's needs. Clients might wish to penalise repeated failure for a particular user with a higher penalty. Both parties might with to limit their financial exposure either by capping the request rate or by varying the price curves depending on the number of requests per second.

5. Research Directions

We are seeking a new way of describing Service Level Agreements between clients and service providers. Changing the form of these agreements should give an incentive for service providers to reduce overheads in the infrastructure. More closely meeting the business needs of the client also provides opportunity for efficiency improvements—if the client no longer pays for a service level which he doesn't need, the service provider need not provide a service level which is unnecessary.

5.1 Specifying agreements

Languages such as Baltic (Bhargavan et al. 2007) can express the interactions between services and make static checks for correctness. Extending such a declaration to incorporate an SLA would mean that this information can then be used to inform the decisions of scheduling software to most appropriately allocate the resources available.

Research into ontologies and the Semantic Web attempt to codify real-world relationships and hierarchies. These approaches might be applied to describing a service levels for different applications.

5.2 Emergent behaviour

Integrating numerous costs and rewards into an agreement may well create complex emergent behaviour. Parties might wish to model check an agreement to determine expected outcomes under different traffic and fault models.

In the simple example agreement mentioned above a service provider might decide to run no services at all overnight due to the small number of expected requests. A client can respond to this strategy by generating synthetic traffic and thus imposing an additional fine on the service provider. However, the service provider can respond in turn by running the service overnight and allowing the synthetic traffic to run up a bill which does not correlate with any expected business revenue.

A model often used in game theory for the expected behaviour of the service provider and the client is that they will always act to maximise their own revenue. In the example above the service provider would not consider switching off the service overnight if the client's response will cause a loss. However, for this reasoning to be valid the client must actually have a possible response available to it.

6. Conclusion

Much research has focussed on reducing the energy consumption of computing itself. However, there are significant reductions to be found when considering the infrastructure which supports computing. Technologies such as migration and automated recovery strategies mean that we could build adaptive datacentres which run closer to the wire and provision their workload according to the available resources.

Unfortunately, current SLAs do not provide much freedom or incentive to do this. We have argued for a more client-centric specification of service level. This in turn gives service providers more freedom to minimize their overheads and helps to ensure that the level of service provided more closely matches the level required by the client's business.

We wish to investigate how these SLAs might be incorporated in languages currently used for specifying the interaction of services within a datacentre and how this information might be used by a scheduling service to optimally provision infrastructure and computing resource.

References

Karthikeyan Bhargavan, Andrew D. Gordon, and Iman Narasamdya. Service combinators for farming virtual machines. Technical Report MSR-TR-2007-165, Microsoft Research, 2007.

George Candia and Armando Fox. Designing for high availability and measurability. In *Proceedings of the 1st Workshop on Evaluating and Architecting System Dependability*, 2001.

Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 273–286, 2005.

Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, , and Andrew Warfield. Remus: High availability via asynchronous virtual machine replication. In *5th USENIX Symposium on Networked Systems Design and Implementation*, 2008.

Andy Hopper and Andrew Rice. Computing for the future of the planet. *Philosophical Transactions of the Royal Society A*, To Appear, 2008.

Jonathan G. Koomey. Estimating total power consumption by servers in the U.S. and the world. February 2007.

David Patterson, Aaron Brown, Pete Broadwell, George Candea, Mike Chen, James Cutler, Patricia Enriquez, Armando Fox, Emre Kcman, Matthew Merzbacher, David Oppenheimer, Naveen Sastry, William Tetzlaff, Jonathan Traupman, and Noah Treuhaft. Recovery oriented computing (ROC): Motivation, definition, techniques, and case studies. Technical Report UCB//CSD-02-1175, UC Berkeley, March 2002.

My Ton and Brian Fortenbury. High performance buildings: Data centers uninterruptible power supplies (UPS). Technical report, Lawrence Berkeley National Laboratory, 2005.

W. Pitt Turner, John H. Seader, and Kenneth G. Brill. Tier classifications define site infrastructure performance. Technical report, The Uptime Institute.

Eric Williams. Energy intensity of computer manufacturing: Hybrid assessment combining process and economic input–output methods. *Environmental Science and Technology*, 38(22):6166–6174, 2004.