Failure is an option

Andrew Rice
13-May-2008

Outline

- Improving the energy efficiency of datacentres
- A declarative datacentre lends itself to automated management
- This allows adaptation – we can run “closer to the wire”
- We need to change service contracts to give incentives to do this

Servers are often underutilised

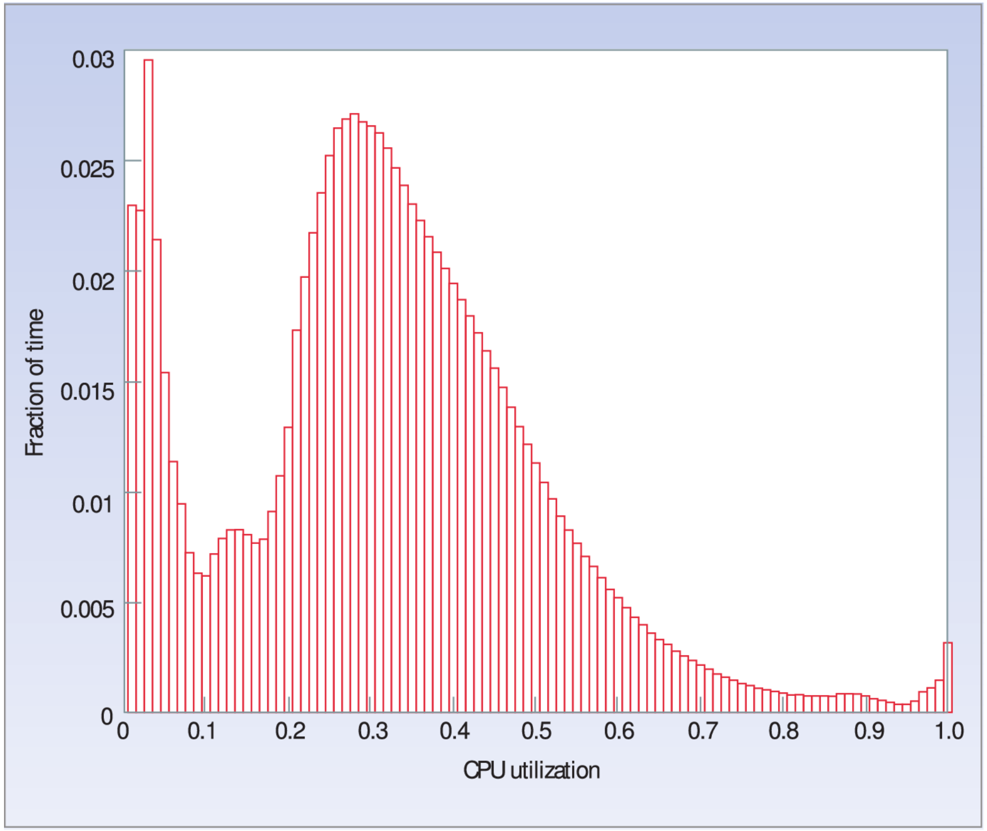


Figure 1. Average CPU utilization of more than 5,000 servers during a six-month period. Servers are rarely completely idle and seldom operate near their maximum utilization, instead operating most of the time at between 10 and 50 percent of their maximum utilization levels.

Consolidation

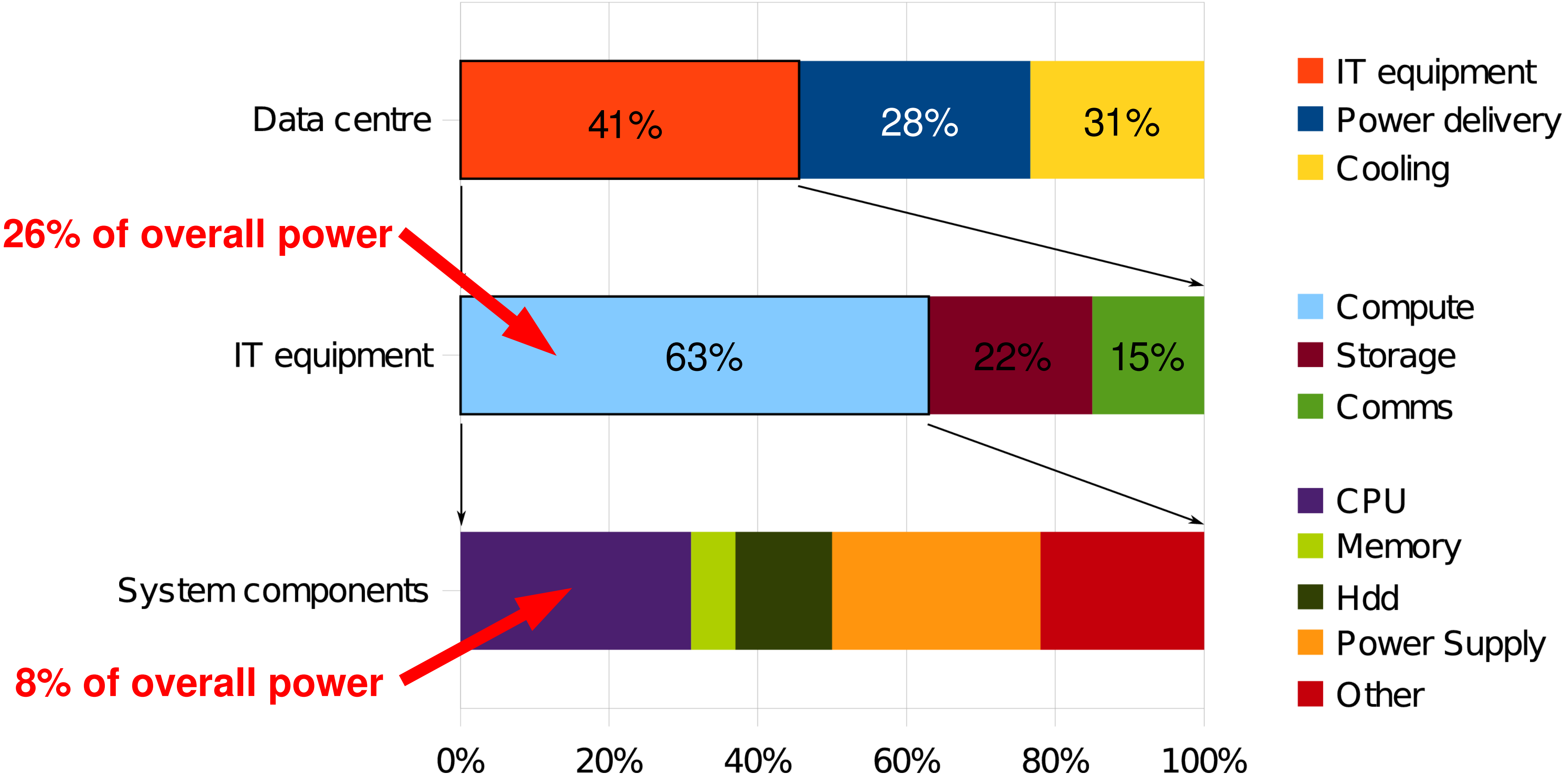
Load concentration

Migration

Description languages

Source: Barroso, L. A. & Hőlzle, U. 2007 The Case for Energy-Proportional Computing. IEEE Computer 40, 33–37. (DOI 10.1109/MC.2007.443.)

But...very little energy actually gets to our servers

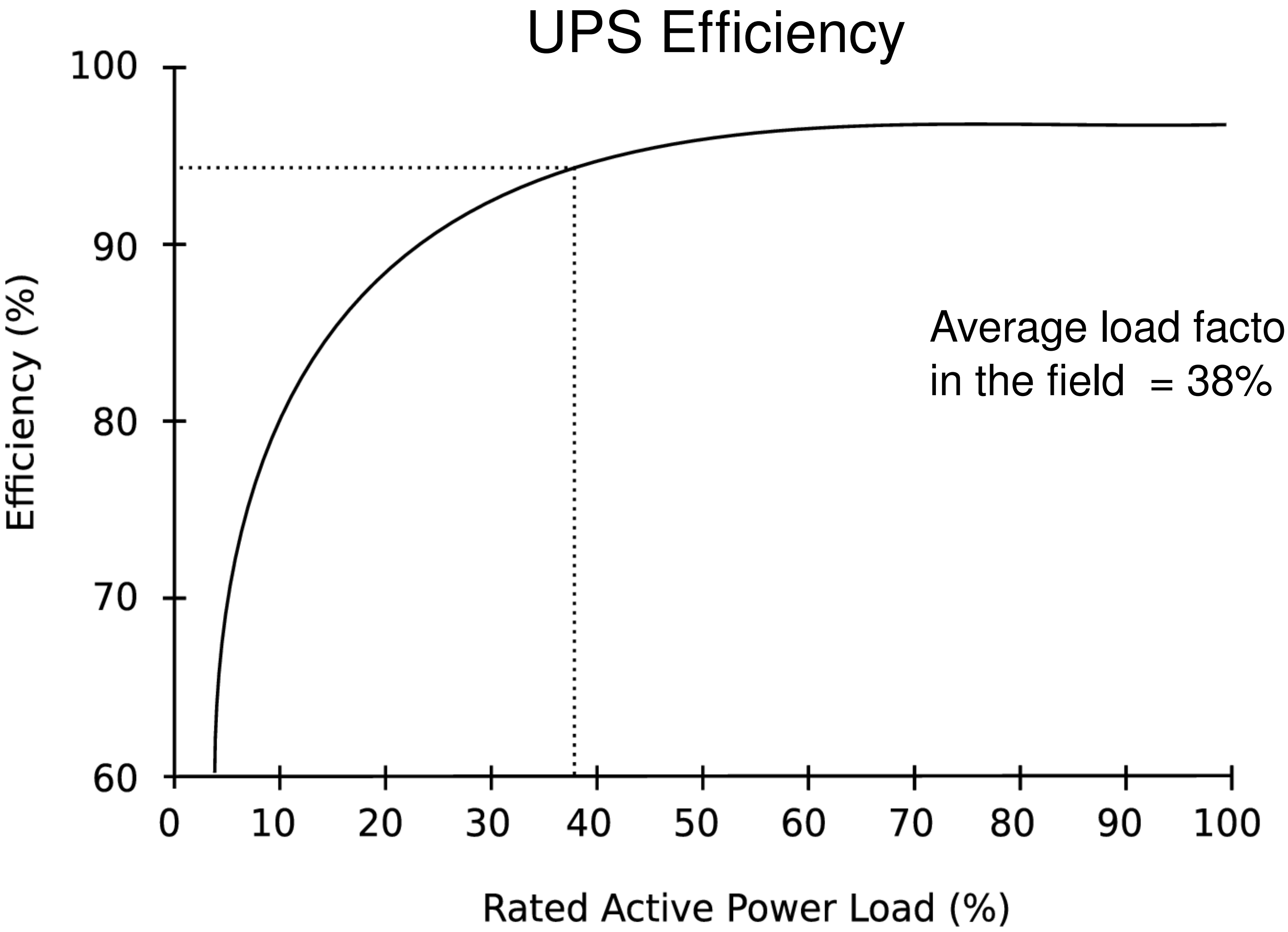


26% of overall power

8% of overall power

Source: Data Center Efficiency in the Scalable Enterprise, Dell Power Solutions, Feb 2007

Energy losses in power delivery

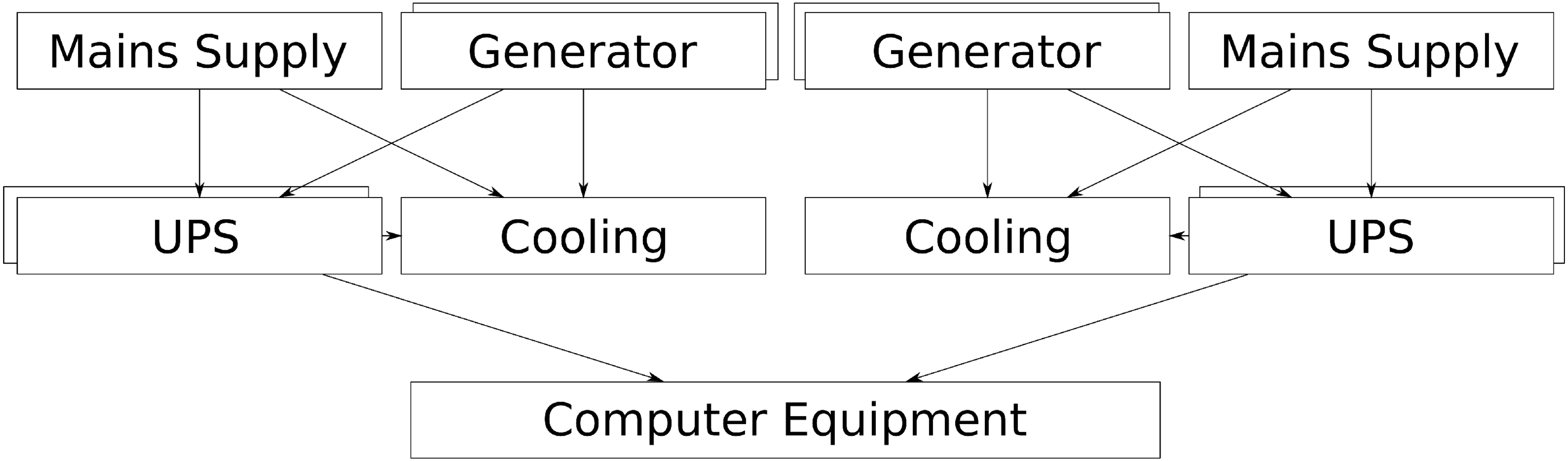


Average load factor of UPS
in the field = 38%

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

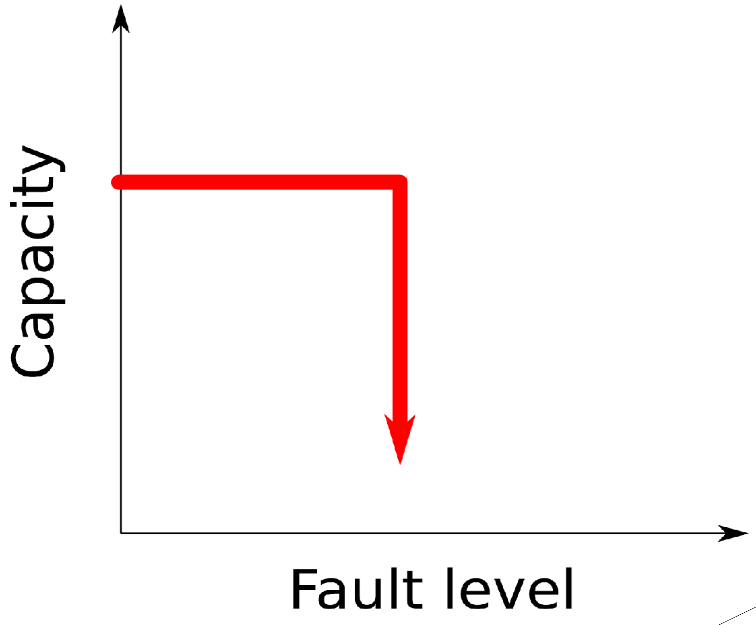
Redundancy causes low utilization

Tier IV datacentre

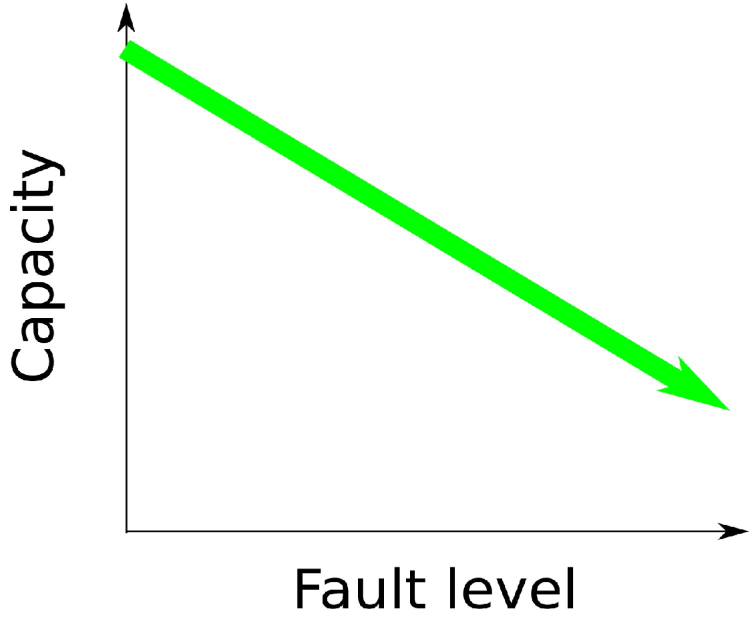


Source:Pitt Turner IV, W., Seader, J. H., Brill, K. G. Tier Classifications Define Site Infrastructure Performance. White Paper. The Uptime Institute.

Maybe we can do without redundancy

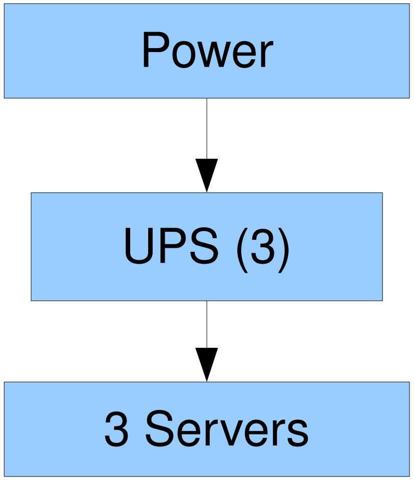


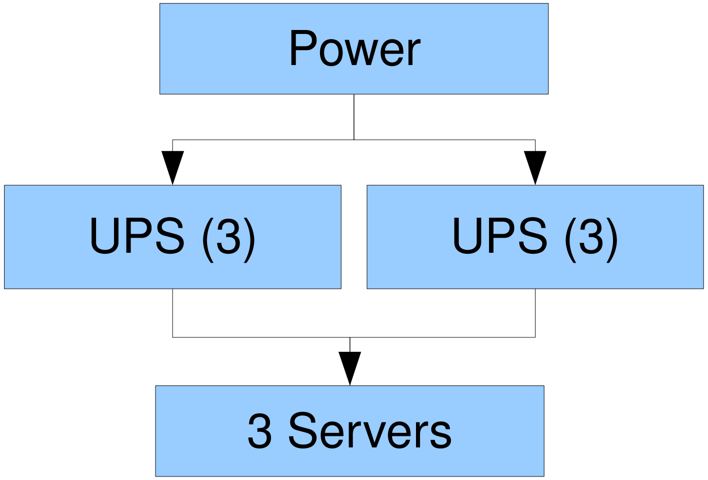
Current approach: failure => disaster

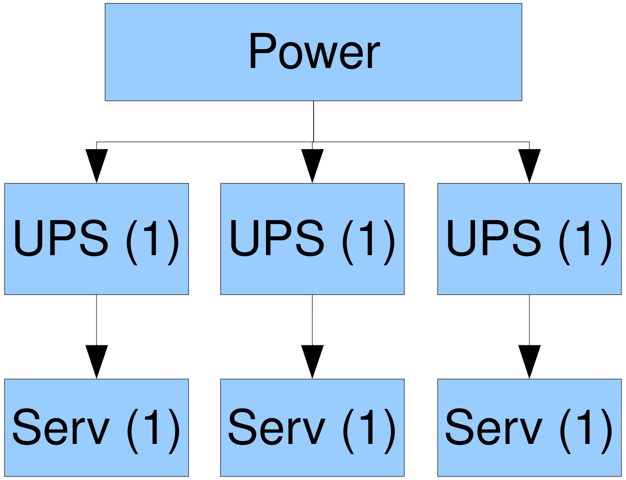


Adaptive approach:
minimize the cost of failures

Ensure that a component fault only partially degrades our capacity







No. Faults	Failure %
0	0
1	100

No. Faults	Failure %
0	0
1	0
2	100

No. Faults	Failure %
0	0
1	33
2	67
3	100

When a fault occurs we adapt to it

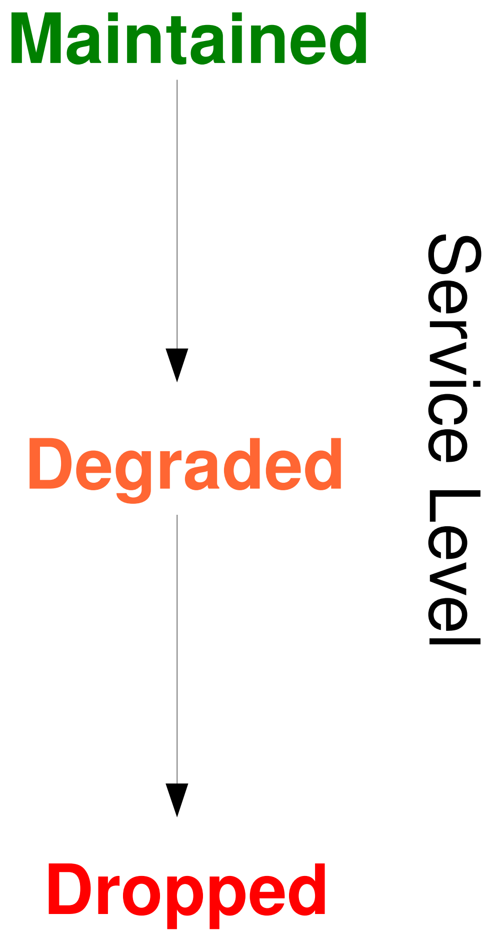
Application-level replication

VM-level replication

VM migration (compaction)

CPU/IO throttling

Stop service



Maintained

Degraded

Dropped

Service Level

The datacentre declaration must inform these tradeoffs

- Service dependencies
- Adaptation possibilities
- Workload models
- Recovery times
- Failure probabilities
- Costs and SLAs

Current Service Level Agreements don't give us much freedom

- Service level objectives are black and white
  - Availability
  - Response time
  - Network uptime
  - Power uptime
  - Scheduled maintenance windows

Current SLAs don't suit the customer very well either

- Its the experience that users of the service get
- The service needs to be working only when its needed
- Response time of a service could be irrelevant compared to network overheads
- Lack of expression in the SLA promotes over provisioning
- Site overload: do we serve everyone badly or limit to a selection of requests?

A payment/fine model for SLAs

- Consider a web service
- Agreement says:
  - For every request serviced the provider gets a payment
  - For every failed request the provider gets a fine
- This is closer to the client's business case
  - Value of a request which succeeds vs. cost of a failure

Research questions

- Economics
  - Price functions, risk management
- Need input from clients
  - Does this actually fit a financial model?
  - What caps and curves are needed?
- Need analysis and forecasting techniques
  - How do we provision and can we make a return?
- Construct end-to-end SLOs?

Conclusion

- Reliability through redundancy consumes power
- We can instead try to adapt to failures when they occur
  - Can still provide high reliability if needed
- Specifying our agreements with clients in a different way could encourage this

Questions or Comments?