The Business Case

Sunday, March 14, 2010
4:13 PM

The business case for clouds

Not at all like the programmer's view.

To a programmer, a cloud is a **convenient way to program**.

To a businessman, a cloud is a way of **reassigning corporate risk** to a third party.

How this course came about

Sunday, March 14, 2010
4:22 PM

How this course came about:
- I was at VEE talking with an IBM VP.
- Computer science has failed IBM now; they don't know where their next VP will come from.
- Most computer science students:
  - Don't understand social context.
  - Don't understand business.
- IBM's Service Science Initiative (SSI): an attempt to put business thinking back into Computer Science.

SSI includes a curriculum for:
- Computer algorithms.
- Business process modeling.
- Social understanding of technology.
- Operations research.

My first attempt: 194SS (Service Science), Spring 2008.
- 6 students, 1 professor, 1 Ph.D. student
- We analyzed a social networking site in Bosnia!
- We concluded that horizontal scaling would not improve performance as much as oodles of memory and memcache(Chiarini and Couch, LISA 2008).

Interlude: sabattical in Norway:)

From now on:

Sunday, March 14, 2010
4:27 PM

For this unit,

Forget that you're a **programmer.**
Envision yourself instead as a **business decision maker.**
Actually, still quite technical, but at a different level.
You won't be programming, but
There is a great need for **strategic architectural decisions.**

The million-dollar questions:

What stuff goes into the cloud?
What stuff stays out of it?
Why?

A small piece of culture

Monday, March 28, 2011
4:38 PM

COBOL:

- Takes a skilled programmer to write
- But it can be verified by an unskilled person, e.g., a manager.

Split between expertise in programming and expertise in business

- The programmer makes it work.
- The manager makes it conform to business needs.

The programmer of the future

Monday, March 28, 2011
4:40 PM

The programmer of the future

- Will not mediate between users and the machine.
- Will mediate between business and the cloud.

Initial argument:

- What's the biggest cost in adopting a technology?

Case study: SIS

- Old system: 10,000 lines of assembler.
- New system: modern code, oracle, etc.
- Major cost: adapting policy to fit.

Lines of responsibility:

- Bursar is responsible for finances.
- Registrar is responsible for matters of record.
- But they have to communicate.

Big lesson:

- How a business operates has incredible inertia
- Changing how it operates is incredibly expensive.

Moral:

- To a business, the main question is not whether a program will work, but whether it will enable or impede policies and process.

Barley's claim

Monday, March 15, 2010
9:40 AM

Stephen Barley:

> Programmers and managers think they can do each others' jobs, but in fact, the two jobs require **radically different skillsets.**
> Programmer: the devil in the details.
> Manager: view from 5000 feet: what are we getting out of it.

Barley's law:

> If you make a change with technology, you never get what you intended. You always get "something else".

Cost and Value

Sunday, March 14, 2010
4:42 PM

Thinking like a businessman

- Value:
  - direct value: sales of products and/or services.
  - indirect value: exposure, market footprint.
- Cost:
  - worker salaries (1).
  - power and air conditioning (2).
  - equipment purchases (3).
- Objective function: net profit
  - = Value (accounts receivable)
  - - Cost (accounts payable).
- **Maximize net profit**, subject to reality constraints.

Economies of scale

Monday, March 15, 2010
9:44 AM

Costs of IT

- Biggest cost of IT: labor (permanent employees)
- Second-largest: power.
- A crossover is predicted within 10 years!
- Equipment cost is negligible, because equipment is purchased infrequently, for relatively little money.

Monday, March 15, 2010
9:48 AM

Risk

One key concern of IT managers: quantifying risks. It might seem that risks are intangible, but they can be assigned monetary penalties.

Kinds of risks

- **Data integrity**: whether data will consist of what you wrote!
  - **Consistency**: do all servers have the same view of the data.
  - **Security**: can people make unauthorized changes to data?
- **Data privacy**: whether unauthorized parties can discover data you agreed to protect.
- **Availability**: whether your servers are accepting requests or not.
- **Response time**: whether a query by a user gets satisfied fast enough.

Some monetary conversions

Data integrity: loss of reputation/customers, lawsuits, government compliance violations.

Data privacy: loss of reputation/customers, loss of competitive advantages.

Availability: loss of customers, work, etc.

Response time: loss of work, sales.

Reassigning Risk

Main reason that clouds exist: **reassigning risk.**

A small business can't completely eliminate the risk of data loss or corruption.

But by using the cloud, you can make it **someone else's problem.**

That someone else can **exploit economy of scale** to do it more cheaply.

Example: AWS

Monday, March 28, 2011
5:07 PM

AWS: Amazon will develop and run your whole site
- without signing it!
- using all of their back-end infrastructure.
- and your branding.

Value proposition: for small to medium-scale business
- You can't afford to develop a site.
- You will make mistakes
- Amazon doesn't make mistakes.
- If you can afford it, it's worth it!

In this lecture,

Monday, March 15, 2010
11:00 AM

In this lecture

Let's concentrate on **availability and response time.**
A service is available if one can make requests to it.
And unavailable if requests aren't permitted.
(independent of how long requests take to complete).

A hard lesson

So far, we've considered absolute measures of response time, e.g., SLAs. These are measures that are **meaningful to us.**

One can save substantive money by redefining availability to suit business needs.

Those are measures that are **only meaningful to business.**

Availability and money

Monday, March 15, 2010
11:02 AM

Availability and money

- Lack of availability costs money!
- People don’t get work done!
- People can't get to your website!
- But how much money?

Cost of downtime

Sunday, March 14, 2010
4:45 PM

Patterson's equation:

Downtime (unavailability) has a **tangible business cost.**

Cost of downtime
= (average revenue lost/hour) * downtime hours
+ (average work lost/hour) * downtime hours.

work lost: things that don't get done while you are paying staff to do them.

revenue lost: sales are not made because resources are not available.

Sources:

Patterson, "A simple model of the cost of downtime", Proc. LISA 2002.

Couch et al, "Toward a cost model for system administration", Proc. LISA 2005.

High availability

High-availability sites are measured in sigmas.

4-sigma: 99.99% up = about 1 hour of downtime a year.

6-sigma: 99.9999% up = about 1/2 minute downtime per year!

Fact: each added sigma costs roughly an order of magnitude (10x) more than the previous one: **cost is geometric in number of sigmas!**

Monday, March 15, 2010
10:02 AM

Understanding 6-sigma

To get 99.9999% availability, you need:

At least one duplicate of your whole infrastructure, ideally geographically distributed.

Ability to instantaneously switch between using copies of the infrastructure.

Hot-swappable hardware in case of failures.

Online backup.

Redundant staff able to step in instantly: 2x to 3x the number of people you need to run the service.

"Doing it yourself costs a lot!"

Exploiting economy of scale

The cost of providing 6-sigma to n organizations **scales sub-linearly,** i.e., less than n times the cost for one organization.

Thus we can exploit **economy of scale.**

One staff can handle **many clients.**

Service-provider's lament

Sunday, March 14, 2010
4:35 PM

Service-provider's lament:

Many customers with **4-sigma sites** want **6-sigma service**.
I.e., they're willing to pay **100x** as much as they need to!
For nothing.

A simple calculation

Monday, March 15, 2010
10:08 AM

A simple calculation:

- 4-sigma site: down for **one hour a year.**
- 6-sigma site: down for **half a minute per year.**
- Worst case: both down at different times: 6-sigma doesn’t' add much downtime.

But what if a 4-sigma site subscribes to 4-sigma service?

- Worst case: down **two hours a year.**
- So you lose **one more hour of work.**
- And you lose **one more hour of revenue.**
- Is this downtime cost worth **paying 100x for the service?**
- Usually, no. **Because you could afford to lose one hour already,** and making your service more efficient than that **wasn't worth it already!**

Monday, March 15, 2010
10:14 AM

How clouds change the equations

- Provide 6-sigma service to everyone.
- Lower prices by serving more customers:
  - Staff grow slowly.
  - Number of machines grows linearly.
  - Power needs grow with number of machines.
- **Crossover: power cost dominates staff cost!**

But…

Monday, March 15, 2010
2:43 PM

But… there are more subtle issues lurking

- 6-sigma is what we can offer.
- It is not necessarily what people need.
- Value of availability depends upon customer perception.
- Simple changes in how we define "available" lead to major cost savings.

Monday, March 15, 2010
11:09 AM

A "power"ful claim

- The simple change of stating your availability in **probabilistic** rather than **absolute terms** can save surprising amounts of **power**.

Rewriting the rules:

- Old SLA: 99.99% uptime.
- New SLA: 99.99% of requests get answered; other 0.01% doesn't get answered.
- These are very different requirements!
- The latter allows you to queue requests and defer them until systems are available.
- So general uptime can be much lower and still comply.
- So costs can be much lower.

Monday, March 15, 2010
11:15 AM

Analysis of the claim

The former claim is that your infrastructure (something big) has to be 99.99% up.

The new claim is that your job queue (something small) has to be 99.99% up.

Obviously, it's easier to make something small highly available!

Rethinking availability

Monday, March 15, 2010
2:27 PM

Availability

- … is in the eye of the beholder.
- … has three axes:
  - Ability to request.
  - When response/ack is received.
  - When operation is carried out.
- Importance of these axes depends upon **business goals.**

True cost of response time

Monday, March 15, 2010
9:53 AM

True cost of response time

Astonishing fact: if a customer is forced to wait more than 5 seconds or so for a search result, **the sale is lost.**

So, if you're going to take more than 5 seconds to respond, **don't even bother**; respond to as many customers as you can in **under 5 seconds.**

Tell the others **"sorry".**

This is called an **admission control strategy** for marketplace searching.

True value of consistency

Monday, March 15, 2010
2:49 PM

True value of consistency

On an Ebay search page,

prices are several minutes old.

On the corresponding item page,

prices are accurate to the second.

Why? There is no inherent value in being precise on the search page. **You can't bid from there!**

Monday, March 15, 2010
2:40 PM

A curious idea of availability

Requests flow in.

You're allowed to reject, say, 1% of all requests.

You must reject anything you reject immediately.

If you accept the request, you must respond within 10 seconds.

Monday, March 15, 2010
2:42 PM

Analysis of the curious approach

- Rejecting n queries is roughly the same as losing n customers.
- So there is value lost in rejection.
- But we have to compare this with the cost of growing our infrastructure to match demand.
- And what we sell has to be available to sell.
- Thus it is not the best strategy to always meet response time goals!