Amazon.com’s Simple Storage Service, Elastic Compute Cloud, and Simple Queue Service: Charlie Bell, Amazon

Charlie Bell, Vice President of Technology at Amazon, spoke on some of the challenges Amazon has faced and the ways in which Amazon solved them.

Charlie Bell: This is our original technology strategy. Application Server sitting on a database. We named them Bert and Ernie. Scaled out over short period, 1998 to late-1999. Very ugly. We took the business logic on the server, wrapped the data, and built out. In 1998, primary customer was retail shopper. Mid-1999, opened Amazon’s technology and website to third parties. Real estate on the Amazon site. We understood customers were one click away from going elsewhere. If we wanted to keep their business, needed to provide product. Carried forward to the NBA Store. Not a single Amazon mark on the site. Began drive at looking for third customer. Looking at developers as customers for Amazon. Target.com site lunched in 2001 on Amazon site. Still operated by Amazon today. No Amazon marks. Totally different look and feel to Amazon’s site.

This is all why we did web services. E-commerce services on Amazon in 2002. November 2004, launched SQS beta. History of web services launch. New ways of looking at infrastructure.

Simple Storage Service. Bucket structure. Few APIs. Stored in two centers. Unlimited as in cloud. Cheap. Direct transfer from storage. S3 concepts. One to 5GB. Not many buckets. Not a great deal of structure. Very simple. User defined key to object. Simple APIs based on standards. Can address every object with URL.

Elastic Compute Cloud. Large and extra large sizes. 4x base and 8x base. EC2 concepts. Point at image, how many you want. Simple Queue Service. Most of you understand queues.

Why Amazon picked these services. Many things done in this environment. Amazon has great deal experience with these patterns and what to build. Area where things keep repeating. Primitive web services. Storage, compute, queuing. These became S3, EC2, and SQS. Services born from our experience building. Operational demands scaled.

One of the patterns, walking through. Application where a client produces files, deposits in S3. Consumer will convert to MPEGs. Picks up job, gets AVI and begins processing. Not too exciting. Things get interesting. Bunch of clients with a bunch of video files. Single consumer, you fall behind. Today you could guess at peak load and provision machines to get enough capacity to handle. In this model when the EC2 gets busy, more EC2. Results in smaller queue. We see this model a lot. Based on needs.

Animoto. New age customer. Built entirely on web services infrastructure. They produce presentations. Give them the images and music. They stitch together in a video by going through, finding the message, creating the video, and shipping it back. Older companies are also using this. NY Times has deep repository of archived images of their newspapers. Thought customers would like to see articles from 1851 through 1922. Took many articles in one format and converted to another. Worked at how to do, then looked at EC2 and S3 for utility computing. Dupe is open source tool that does what mapper does. Stitched together a basic workflow. Able to do in five days.

Operations. Heavy lifting of web services. We see all the time. Can’t predict what workflow of application will be. Either too little, expediting everything, or too much with excess capacity. S3 objects stored went from 800 million to 5 billion. Peaks and valleys become more visible. Growth rate of transactions per second on S3 went from 16,607 in April 2007 to 27,601 in October 2007.

What went wrong. Nothing is safe. Data centers can be hit by wind storms, power outages. Build redundancy and something goes wrong that can’t be controlled. Frog DNA is what took dinosaurs down in Jurassic Park. Unexpected technology.

We’ve all seen undergoing maintenance screens. 365 Main, a data center in San Francisco. Had unusual problem. Lost redundant power, generators. Redundancy failed. Many sites went down. Infrastructure fails. Stuff happens. You guys all know you can buy most expensive rated storage, computer, and it still finds way to fail. Businesses not public still have issues. When they die, you don’t get them back the same way. Internet connections drop, can’t get network.

Everything fails, all the time. Routine failures. Sometimes they fail, taking out large quantities of infrastructure. Take out a machine or a rack. You’re ok if you did right. But a whole data center with multiple companies infrastructures. In San Francisco, question was how quickly they could get back up. Larger the footprint, harder the recovery.

Why use web services for S3 and EC2 and SQS. Probabilistic. Physics. If you use enough unstable machine, you can come up with probability of failures. Losing a system.

Service Level Agreements (SLA).

With this scale and reliability, we can offer SLAs. I’ve looked at web services. I’ve looked at companies who claim to have SLAs and really don’t. In end, do they work or not? Look at public data to determine.

One more customer. Blue Origin. Little rocket company. Private company, private funded, working on space vehicle. Video of a capsule taking off and landing. Had panic. When they posted video they knew would get a lot of uptake. They knew they couldn’t handle. Via Amazon, put video up. From about 700 applications, peaking at 3.5M requests. Transferred 758GB of data. These guys pre-provisioned enough machine capacity and bandwidth deals, virtually impossible. They made decision a few days before launch, to do this. Off running.

That’s about it. Just wanted to give you a flavor of our services and our operational challenges. Q&A.

**Participant: Commend you. Extraordinarily good job. Talk about …**

Charlie Bell: Question was about Mechanical Turk. Mechanical Turk is an 18th century hoax that was produced touring the Crown Heads of Europe. Clock Work. Everyone believed you could do anything with. Someone built chess playing machine to show off to Crown Heads, saying ran with Clock Work. Turns out there was a very tiny Grand Master in it. Beat Benjamin Franklin and others before hoax discovered. Today, Mechanical Turks farm out information, re-gathers, and comes back with summary or results. Primitive. Used for huge variety of applications. Fossi search, Jim Grey search. Both lost. Opportunity to use satellite imaging. Human can look at and determine if airplane in image. Applications in that vein. Applications that recognize this address matching a picture. Language translation. Any semantic data. Different kinds of applications. Jim Grey search. He was lost off San Francisco Bay area in boat. Massive search. Companies getting together to search. Lots of work put in. Fascinating technology. Wonderful example of web services in toolbox, looking at in totally different ways. Has some of the same scaling characteristics. Mechanical Turks have broad abilities. See classic applications emerge. One of the poster children within operations at Amazon. Doing very well.

**Participant: …**

Charlie Bell: Hinted at in talk. I’ve been at other end quite a bit. We provide SLA. Growth high. Based on Amazon launching SLA, have we seen lots of growth in service? Impossible to tell. Second question, how we should think about SLAs. We all have suppliers, might have outsourcing deals, rely on folks. At end of day, SLA is post mortem. Small player offers fabulous SLA. Turns out they only perform post mortem’s. What matters is history. Do they have operational experience. Experience in my application. Availability and uptime and how long to put 4GB object in versus 20KB, over Internet. EC2, firing Linux instances. You own the OS. Reliability and performance at operational level, dictated by what you do. Best way to figure out is to just go use. A customer wanted to try us out. Asked for a free trial. Gave the guy a dollar and told him that was worth ten free hours. Cheap to do and offer. Early adopters running applications to understand how it will behave for them. Best thing to do is try it and also look at public history. Not a lot to compare with today. We have good reliability record. Somewhat unique and therefore hard to compare. We’ll see more entries into this space for you to compare with.

**Participant: Amazon.com, web services …**

Charlie Bell: Plenty of database applications that don’t user S3. When you get to use cases that fit services, high percentage. We used before we made publicly available. Incarnations before that. For use cases represented, high percentage.

**Participant: You started with three offerings to see how far it would go. Retrospect, have you found things you can’t do with those three? Working on new offerings to fill?**

Charlie Bell: We saw these three fit our applications. Are there applications these don’t satisfy? Yes. Full space of different application patterns, kinds of things as you fill out infrastructure that can be added. Interesting in that the mantra of Internet and web services is keep it simple. We find with these primitives, people can do lots of things. You’d think they wouldn’t work. Running databases on S3 as a file storage. I thought would require API. In answer, yes, looking at things we might need, but interesting how far these three components actually go.

**Participant: What are you using behind … AMI?**

Charlie Bell: Technology with EC2 and plans to support Windows. We’re agnostic. I suppose we could figure how to run Windows in there. Customers tell us what they want and they want to run Windows. Amazon keeps trying to give clients what they want. Need to solve licensing. Fundamentally, we’re not biased. You can try anything you want in your S3.

**Participant: S3. Lots of objects that don’t change. Optimization …**

Charlie Bell: Question is, given data access pattern, are we looking at keeping lower level of redundancy on these objects. Durability is something. Number of times of access doesn’t matter. We don’t talk about how we do storage underneath. Where that will show up is in price. We think this price, understanding market and what people pay, this price is competitive for what you want. Think about issues with near line solutions. Database backup strategy. People using S3 for backup. Stream in parallel right out of there. No tape searches or data recovery. Near term people look at as fitting. Comes down to price and alternatives. Our price is good.

**Participant: Energy consumption …**

Charlie Bell: Something that is interesting. Energy. Public good in not needing much. incentive to use less. Moore’s law drops compute, but not power. Do everything to reduce amount of power consumption to save money and protect environment.