Facebook at Pacific Crest Global Technology Leadership Forum

Company Participants

- Evan Wilson, Analyst
- Jason Taylor, Director, Infrastructure Foundation

Presentation

Evan Wilson {BIO 6804034 <GO>}

Hi, everyone. Thanks for coming. I'm Evan Wilson; I am the senior analyst following Internet and games at Pacific Crest. Very excited to have Facebook here presenting to all of us today.

We have Jason Taylor, who is the Director of the Infrastructure Foundation at Facebook where he leads the groups that manage hardware, design, supply chain, technical program management, server budget and allocation and the long-term infrastructure plan.

Prior to joining Facebook in 2009, Jason worked at a number of startups including BlueMountain.com. And he also holds a PhD from MIT in Ultrafast Lasers and Quantum Computing. So everybody, Jason Taylor.

Jason Taylor {BIO 18251157 <GO>}

Hello. Okay. So again my name is Jason Taylor. And I lead the infrastructure groups responsible for servers with Facebook.

It has been a pretty exciting last few years in terms of scaling up Facebook. And what I want to do now is take the opportunity to share a picture of our infrastructure, with a view of what we think is coming next. And I am here to talk about places where technology can go.

It could create efficiency opportunities industrywide and potentially disrupt existing markets over the next two to five years. So my focus will be mostly on technology and the potential of technology over the medium to long term. But definitely over the next few years.

I will begin by reviewing Facebook infrastructure; talk about efficiency so far as Facebook, what areas we have been pursuing; talk about something -- an idea that we are working on right now, which is disaggregated rack. And I will go through that in detail; and then talk about some new components that we think could be

designed and brought to market over the next couple of years that could be highly impactful in both the data center space and just IT broadly.

In terms of scale, we have 1.15 billion users. 84% of our monthly active users are outside the United States. And we have data centers in five regions.

Again, in terms of stats, lots of users. 700 million people use Facebook daily.

350 million photos are added each day. We're at about 240 billion photos aggregated so far.

And about 4.5 billion Likes, posts. And comments per day. So it is a very active and a very large infrastructure.

In terms of cost, our infrastructure spend in 2012 was $1.24 billion for capital expenditures related to the purchase of servers, networking equipment, storage, infrastructure. And the construction of data centers. And at that scale, efficiency work has really been a top priority for Facebook for several years.

I am going to talk a little bit about Facebook's architecture, give to you an idea of what we think of when we think of infrastructure at scale. And how we think of computing. So a couple of things I will talk about.

One is cluster. When I say cluster, I mean network cluster. So that could be a two-post cluster, a four-post cluster. But it is essentially a group of servers that have lots of networking capacity between them and they are organized as a group.

And we use them in really three different ways. The first is our front-end cluster. For front-end clusters, this is really the workhorse of Facebook. Any time you do anything on Facebook, post a message, look up something on your phone, anything in that direction, a request is sent to one of our front-end Web servers; and then those front-end Web servers work with the rest of the servers at Facebook infrastructure to deliver the content.

Now within those front-end Web servers there are some services that really scale directly with just how busy the site is. So we scale Web servers based on how many requests we get per second and the composition of those requests.

So ads, multifeed. And some other small services tend to scale with that. And so we put in that unit of scaling, which is our front-end server. We have lots of front-end clusters.

In terms of ads, these are racks and racks of servers -- and when I talk about racks, I am talking about what we think of as our unit of capacity, which is a server rack. It is about 8 foot tall, it is about 20 inches wide, 26 inches deep. And it contains anywhere from 20 to 40 servers.

Now multifeed is something I will get into quite a bit. But this is essentially newsfeed on Facebook. So if you think of the Facebook page and you think of that center column of recent activity, we have a service that is dedicated to just serving that piece up. And I will talk about that software in a little bit.

We also have service clusters. These are essentially computers that are dedicated to serve a particular purpose. So search has its own set of computers; photos; messages; and several others.

Then back-end clusters are all about the databases. So with the databases, has everything to do with being able to run the databases, back them up and keep them reliable.

Our first push for cost optimization in hardware was really the genesis of Open Compute. In 2010 we built what we call vanity-free servers which is -- they are really standard servers except we have removed all of the components that are typically found on servers that we don't use. We also co-optimize the data center and the computers together.

I will talk a little bit about what we have done with data centers. But the idea was very much to solve the engineering problem of how do you efficiently put a server online in a data center? Not, how do you built a computer that a large variety of customers would want to use? So that is Open Compute; it is all focused on Web and large infrastructure scale.

Now getting into some of the designs of some of our services. This newsfeed rack -- it is a very common design for us. It is called a leaf aggregator. What happens here is within a rack of about 40 servers, all of the servers cooperate to answer a single question.

So what happens is that you have queries come in and it hits an aggregator; and an aggregator is just a piece of software that is running on all the servers. Then there is also another piece of software called a leaf which is running on the servers as well; and the leaves take up most of the RAM and the aggregators do most of the computation. That is the trade-off.

The reason that this is, is because the aggregators will be asked -- okay, Jason is viewing the site; and the Web server says here is a list of all of Jason's friends; tell me what they have been up too. In each one of these racks is an index of all of the activity of every user on Facebook for the last couple of days.

So what this is, is this aggregator will receive the request and then ask all of the other servers in the same rack, it will ask all of the leaves -- do you have any data for user, for this set of users and this set of users and this set of users? Then that subset will be sent to the aggregator, the data will be reduced and then sent off to the Web.

So when we say that a rack is our unit of capacity, no one of these servers could answer the question properly alone and they all do have to work together. This is really just out of scale. We needed to put terabytes of RAM online and make it accessible in order to just serve this up. And so we needed to adopt something that is cooperative in this way.

Talking through a little bit about how Facebook serves our clients, what we do is -- I have drawn time going down. So when a request starts, it hits a Web server. Then the first thing that the Web servers do is that it checks login, authentication; it hits a lot of caching servers. These are memcached servers.

Then later it might hit a feed aggregator, like a multifeed or this newsfeed I just described. Then that is just a list of stories. There is no data there; it is just indexes.

So then you might hit cache. We have a very high cache hit rate, maybe 96%. And each one of those memcached servers is probably 1,200 servers per front-end cluster. So really lots and lots of cache servers.

You might need to look up something from your database; you might need to add that data to memcache; pull up some ads; and then really send out the data. Now, things are a little bit more complicated than this. But this is essentially how a hit lives on Facebook.

What you will see is that this is really data-intense. Right? We pull up a lot of data in order to serve any page on Facebook. If you just look at it and think of all of the different pieces of data, it is really a lot of information.

Now in terms of how we built out Facebook, one of our big wins or one of the things that has really helped us a lot over the last few years is in our infrastructure we have only five server types. So at Facebook, an engineer can have any server they want as long as it is one of these types. That's it. There is no variance; there's no changes.

So what we have done is we have designed each SKU with a particular major service in mind. So the Web, we have tons and tons of Web servers; and Web servers are all about putting a lot of computation online very cheaply.

Databases are all about providing enough IOPS in order to service the request coming from the Web servers. Hadoop is kind of our Big Data service. So we need a lot of compute and a lot of storage.

Photos are all about the most gigs for the least number of dollars. So it is all about storing lots and lots of photos, lots and lots of data, petabytes of data.

Then feed is that one which is -- type VI is feed server. This rack is very popular. It is that leaf aggregator I talked about. And a lot of services consume it.

So the major advantages to having a constrained number of servers -- and this is standardization -- it is that we get volume pricing, right? If we are negotiating for only five different types we can really focus on the negotiation. And we can really work to make sure that we are getting the best deal for Facebook.

The other advantage, which is not obvious, is repurposing. With repurposing -- when services launch they are launching to 1 billion users. That is a lot of users, right?

And even within a service launch some products could be more or less successful or more or less popular then another one. But we have to make sure that we have enough servers online to catch all of them.

So rather than essentially buying a different type of server for each service and then buying the maximum amount they could possibly need, we run it I'd actually say a little bit more like a mutual fund, where you might have five or 10 services launching, we might have something -- new services coming up at any time. And we just kind of make a good bet that says -- well, this guy, this guy. And this guy, they are all going to be very popular; but we are pretty sure at least two aren't going to be quite as big as projections. So we need to slosh about around.

By having identical servers across all three services we can repurpose. So that is a really big win for us.

It also makes for easier operations. Essentially if every server is not new then it is very easy to build and maintain and you get very good at it over time.

And because we have a reduced number of servers we can also allocate servers in hours rather than months. So we always have some of each of those five types on hand for any unexpected need.

Now the major drawback is that we have 40 major services and we also have 200 minor ones. So not everyone fits those five types perfectly.

Also, service needs change over time. Disaggregated rack is one of our ideas, one of the things that we are working towards that we think can address some of those problems for really us and think it would be useful industrywide. We will talk a little bit about that in a few minutes.

So getting onto efficiency. And this is a short section where I just want to summarize where our focus on efficiency has been for the last few years. And then get into a couple of ideas that we think are going to be meaningful for the future.

In terms of data centers, with data centers, traditional data centers, they are like power-dense Walmarts, right? They are just -- it is a ton of power; and you have got air conditioners. And they tend to be really cold because servers consume a lot of

power. And then you put them way too much air conditioning capacity. Essentially data centers are really energetically inefficient.

The way that you measure that in the industry is PUE. PUE is the total power consumed by a data center at the street, divided by the amount of power consumed at the server level.

So a beautifully, perfectly efficient data center where all of the power is going into the servers would have a PUE of 1. Typical data centers and the majority of data centers out there have a PUE of about 1.9, which means that your electricity bill, you are paying about 90% more for every watt of electricity consumed by the server. You're paying about 90% more just to air condition the room.

So that is a very inefficient system. And if you start being a little bit clever you can do hot aisle containment, cold aisle containment; there is lots of things that can get you down to about a PUE of 1.5. But the real trick is to get rid of all of the air conditioners completely and to do a lot of work on the electrical system and the UPS system.

We have talked about this before. But we are very proud that our PUE at our data centers is at about 1.07, which means that we are really just pumping cold air from outside, running it across the servers, taking the hot air and just pumping it outside the building. You can't really get more efficient than just running some big wall fans. So we are very happy with that. And until the data center team figures out a way to generate power, that number is not going to go any lower.

So the servers, again we focused on vanity-free. So just building a server that was exactly what we needed and really not much more. And then a lot of supply chain optimization. So in building our own servers we can get much better deals on commodities, things like that.

On software, we spent a lot of work on building a more efficient Web server. That is a more efficient PHP Web server stack. And that is our HHVM effort. And that is something that you can read about that is very interesting. The cache and database tiers we work a lot at.

Then Web and service optimizations. Efficiency at Facebook is shared across all of the major product groups. And they are the best who are -- those guys are most able to optimize their code. So really those men and women are fantastic. And they understand exactly what they need. And so we work very closely with them on efficiency.

Getting on to the next opportunities, really we are talking about disaggregated rack. When we talk about disaggregated rack, we are talking about being able to extend the useful life of the components that a server is made of. So extending the useful life of compute, RAM, drives, flash, all of those sorts of things. And I will get into those details.

Then also, we think that essentially the components that we work with today are the result of really 20 years of the industry focus on providing solutions for desktop computers. If you actually look at large scale, if you look at large infrastructure scale, you would probably come up with very different types of components. And I will start to talk about what those components might look like here.

In disaggregated rack, if you ignore the fact that there's 40 servers and you just think of what we're actually spending money on -- like what is the resource that's online? The resource is compute. So 80 processors worth of compute; about 6 terabytes of RAM; 80 TB of storage; and in some cases, 30 TB of flash. So those are the raw resources that we need to connect within the rack.

So if you think about not the server and you think about the rack, you really just want to get those resources on line. And you want to be able to have an architecture that can fit for that. So when we think about these we're really thinking about sleds that look like servers that are just providing one resource. Right?

And all of this focus is on disaggregated rack because when you build something that is not very simple -- like, our Web servers are really simple. It is just compute; it is all about just mowing through lots of computation.

When you think of a database you have got multiple bottlenecks to manage. And you are always buying for the weakest link. So if your database is out of storage, you are buying more databases; if your database is not functioning well, you are buying more RAM. Essentially there is a lot of waste in not hitting those resource allocations perfectly.

So with compute, we really think that a standard server is pretty close to what we need. It is really just bringing on as many cores as possible for as cheap as possible.

For RAM we think of basically an idea of a RAM sled. So a server that just responds with key-value pairs, a server that does nothing more than fill that leaf thing of -- do you have this data? Yes I do; here it is.

It is nothing more than that. It is a very low power, probably very, very modest processor there.

When you think of storage, we like our Knox sled. And really that is all about just bringing on compute. We think we can build a small server that you can slam into the back of a Knox sled and just basically put those four -- those 15 drives directly onto the network.

Flash. Flash is really -- it is a sled, it is an appliance. It is really just about putting a lot of flash on a very high network interconnect.

So what we can do with these building blocks is we really have three wins for disaggregated rack. The first is the server/service fit across services. So we can make a rack that uses only maybe four commodities, four basic sleds. But fits a lot of services much better. The service/server fit over time; I will talk about that a little bit. Then really the key is a longer useful life.

So if you think of multifeed, you have a type 6 server. This is essentially the amount of resource that a server needs in both RAM and compute; and in the box is how much the server has been provisioned. So the service need is the squares; and then the overall box is what the server can do.

Now if you look at another service, one that is more RAM-intense and less compute-intense, we have essentially wasted this CPU resource. CPUs are expensive. And if we were deploying services that don't use all the CPU effectively, then we will be buying for RAM, which is not the best thing to buy for. We can spend a lot of money there.

Over time, products change, goals of services change, databases evolve, you get more data per compute -- all those sorts of things. So over time you might find that you don't have enough RAM. And with disaggregated rack you can just slam in another RAM sled and then the rack has more RAM in it; it's a much more serviceable, scalable way of building computers.

The other key is useful life. Typically servers in data centers are for about three years; sometimes a little bit longer. But generally that is about where a lot of them come off-line. And it is because at three years they become really awkward to code for.

Because if you are building a service you are really building -- you want to be building for the future, where these resources are going, not where they were. And if you have a server that is three years old, it is very difficult to do that.

But if you think on the rack level, you can imagine keeping compute for maybe three to six years; RAM sleds for five years or more; discs are easily four to five years. Then flash could be six years or more; it is just when the flash wears out.

So if we bring this back to graph search. And you think about -- well, we've got so much compute, RAM, storage. And flash, if in a year from now the graph search team needs just to index more data -- which is pretty likely considering Facebook data growth is pretty significant -- it would be much better for us to just slam in a few flash sleds then just buy 50% more racks. So that is essentially where one of those service-fit-over-time wins happen.

So again, we have maintained volume pricing and serviceability. We can essentially provide custom configurations for each of these services. So if you've got a data center full of these kinds of servers, everybody can get a slightly different mix of racks.

The hardware can evolve with the service over time. That also allows you to extend the lifetime, which is very helpful.

You get the smarter technology refreshes, which is again better useful life, better depreciation.

Then speed of innovation is a nonobvious one that we think is important, which is essentially, by having a structure like this we can try out newer technologies much more easily -- essentially compete one compute resource against another. Now physical changes are required. And there is an interface overhead.

But when you look at the approximate win estimates -- now these are numbers from, I think, Fry's Electronics. So just looking at really commodity pricing, what IT pays for. The point is that if you are conservative with all of your assumptions, disagg rack could give a 12% to 20% OpEx savings. And with OpEx I'm talking about inclusive of depreciation. So it is really a TCO win. And more aggressive assumptions promise between 14% and 30% OpEx savings.

Now this is different for each infrastructure; I am not forecasting exactly what this win is going to be for Facebook. But this is the size of the win that feels appropriate for a good investment on our part and also industrywide to be very interested in.

Now when we move onto new components. And if you look at the last 20 years, it is really 20 years of the same design. The computers that we use in data centers are almost identical to a 386 desktop from 20 years ago.

They all have -- in fact I think that one had a 3.5 inch hard drive. They are all basically just a computer, PCI, eBUS.

And if you look at the innovations over the last 20 years, they have been significant. But they really amount to a math co-processor. We can now put two processors on every server; that was a nice win. Multicore processors allowed chips to scale, which was significant.

GPUs are great at vector math. And that was a nice innovation for anything that uses graphics or image processing. And flash memory is, of course, transformative for lots of database workloads.

Now, when you think about the fundamental components of compute, RAM, disk. And NIC, which have always asked for more but we have never asked for anything different. And what I am going to talk about is ways to pose an engineering problem for each one of these, where the fundamental components that these manufacturers can bring to market could be very different and scale much better for the data center and for the infrastructure of scale markets.

Today, server CPUs are you really just compute-dense version of desktop servers. What we see is really mobile phones are driving the development of lower-power processors.

And one of the neat tricks that they are applying is system-on-a-chip which is, you have a processor and there is a lot of peripherals that need to support that processor. And they are just putting that onto the same silicon. In some cases, the same package.

But it is a really nice cost win, it is a really nice power win, real estate win. And we think that the server market and essentially server processors will be adopting system-on-a-chip approach over the next few years. It is a very powerful idea and we think it is very interesting.

Now when you think of RAM, today the DDR standard, which is what all RAM components are built for, is coevolved with the RAM. So RAM fits it really well.

We think that there is a lot of alternative memory technologies that really couldn't hit the DDR standard in terms of a cost per performance basis. But because of the way computers are designed today, they don't have an entry point into the market.

There is no place where they could actually start because there is no on-ramp. There is nothing that will connect them.

What we think is that a processor architecture, a software architecture that is really focused on near RAM and far RAM -- essentially you've got really fast DDR stuff and then you have got perhaps or maybe even an alternative technology that is faster than DDR; and then you have got much slower things. So it is really just filling in the space between RAM performance, which is on the nanosecond scale. And flash performance, which is on the microsecond scale.

Somewhere in between there, there is really a lot of interesting technologies that aren't -- essentially they are not ready, in part because of the way we design software. So we think that this is actually a very, very interesting opportunity for the next few years, is figuring out a way to allow more of those alternative RAMs to come online.

In terms of storage -- so photo and data storage uses 3.5-inch drives almost exclusively. That is the workhorse for photo and data.

They are fantastic, the technology is great, everything works. But it is also designed for -- in somewhat desktops, it is also designed for workloads, where you have to do a lot of updates.

What we see at Facebook and what we think we're going to see more and more of in the industry is of course there is Big Data and of course there is massive, massive

amounts of media and photo storage. And that is not going to go away. Projections put us at going from about 3 zetabytes of data, which is 1 billion terabytes -- that is 1 billion hard drives from a few years ago; it is amazing -- out to about 40 zetabytes of data in 2020.

So we are going to see lots and lots of data that needs to be stored. I think the insight that I would like to share is that a lot of that data is really cold. In other words, nobody ever looks at it.

Photos from several years ago people don't often view. Essentially there is data that you want to keep around for a long time; you're not sure of the value but you need to keep it because it might be valuable in the future.

The key thing that you want to understand about all of this is that you don't change these. You don't edit a photo later. You don't change something about the photo. Once it is written, it stays forever.

Once data is collected you don't want that data to decay or change over time. So what we actually think is that flash and flash alternatives, potentially lots of solid-state technologies, could deliver ultra-low write entrance.

So the whole flash market has been really geared toward and pushing towards higher and higher IOPS, higher and higher performance, higher and higher endurance, really pushing that edge of performance. And we're actually saying that if you go really just to the lowest common denominator, build something that kind of barely works -- build something that you can write to it once. And if you try to write to it again you will ruin it.

Because there is a whole -- there is a ton of storage and a ton of workload where you really just want to have the data, keep it for a little while, or keep it for a really long time and never touch it again. We think that is a big opportunity and we actually feel like solid-state could be very interesting.

Of course, hard drives can absolutely scale, as can other maybe optical drives, things like that. I've got two minutes left. Okay.

So in terms of flash, maybe we can just talk about this just a little bit more. Flash is used in databases. And again we have been focused on higher and higher write endurance and performance.

And I think that the idea of WORM flash -- we can remember back to WORM drives a long time ago. Essentially Write Only Memory, things that -- where you write it once, you can't change it again. We think that this is actually a really interesting opportunity for flash.

We think that perhaps flash, perhaps other solid-state medias. But essentially that is a very different engineering problem to present to a lot of these fantastic engineers at these major companies.

Just one sanity check. If you think of a rack of flash versus a rack of drives, a rack of drives is about 2 petabytes of data, a rack of flash is about 4 petabytes of data. They consume about the same power, they weigh about the same amount.

So you can get that density online. And this is using laptop drives. This is a ridiculous form factor.

You build it differently. You focus on the silicon. You build far, far denser silicon modules. And you could really create a really interesting rack of storage. We think that is pretty compelling.

Last word is on NIC. So 10 gigs is really common in servers today. We have been on it for a couple of years.

We feel like essentially over the last few years there has been a lot of really interesting research and a lot of really interesting work done on optical NICs. Right now most forecasts predict 40 gigs in 2018. And 100 gigs in 2020. And this is the threshold at which it becomes somewhat commoditized, essentially very cheap.

We think that that can actually be probably pulled in by a few years. We think over the next two to three years we will probably see cost-effective, maybe even 100-gig NICs on the server level. And that actually is a very exciting thing for disaggregated rack, flash. And a number of complications.

So really I think we're at an exciting time in technology. I think there is a lot of interesting options that are going to be developed over the next few years.

And if things go well, we think that these changes can provide efficiency wins for not only Facebook but really the Web scale and large-scale infrastructure as a whole. So thank you for your time.