Credit Suisse Technology Conference

Company Participants

- Jason Taylor, Director
- Stephen Ju, Analyst

Presentation

Stephen Ju {BIO 6658298 <GO>}

Good morning, everybody. I am Stephen Ju, Internet equity research analyst here at Credit Suisse. I am joined on the stage by Jason Taylor, who is the Director of Infrastructure at Facebook. Jason leads a group that manages server budget and allocation, designs hardware, performs architecture reviews and curates the long-term infrastructure plan for the Company.

Jason holds a PhD from MIT in ultrafast lasers and quantum computing and a BE from Vanderbilt in physics, electrical engineering and math. So without further ado, Jason, take it away.

Jason Taylor {BIO 18251157 <GO>}

Great, thank you. So again, I am Jason Taylor. And today I want to talk to you about a few things that we are doing with our infrastructure, give you a view of our current infrastructure and then talk about a few developing technologies that we think could provide efficiency wins industrywide over the next two to five years. And we will get into it.

So we will do a review of Facebook scale, talk about efficiency, talk about an idea that we are working on right now called disaggregated rack. And then get into some new components that we think will be very interesting.

So 84% of monthly active users are outside the United States. We have data centers in five regions, we have a very sizable infrastructure. It is also very busy infrastructure. So 1.89 billion users, monthly active users, 728 million daily active users. Those users upload 350 million photos per day. And we have over 240 billion photos. In terms of activity, 4.5 billion likes, post and comments. So it's very busy.

In 2012, we spent $1.24 billion on capital expenditures related to the purchase of servers, networking equipment, storage and the construction of data centers. So at that scale, efficiency has been a high priority for us for several years.

Now I want to talk a little bit about our infrastructure. Now here, we have three different sections. One is called a front-end cluster, a service cluster and a backend cluster. That cluster refers to a cluster of servers. And that is the same as the network cluster. So we think a lot about the network. We do tons of bandwidth inside of our data centers. And so we have to be always mindful of that.

When I talk about a rack, a rack is -- it's a rack of computers. It is about 8 foot tall, has about 40 servers per rack. And this front-end cluster is really our window to the world. So everybody who comes to Facebook, all of the hits that come through Facebook, they all go through a front-end cluster. Then the other clusters support that front-end cluster, that front-end Web server.

So to translate that to servers, there is about 10,000 Web servers per front-end cluster and about 144 terabytes of cache spread out around about -- over around 1000 servers. We also have ads racks in there. So these are servers that are dedicated to serving ads. Multifeed, which is that center news column of Facebook. It is kind of the main page of Facebook. And all of these work together to generate a page. This is what they look like.

So we talk about vanity-free servers. And when we say vanity-free what we mean is that we have removed the components that are really not necessary for operation at scale. So no VGA ports, USB, all of that kind of stuff. It is really not necessary. So we cleaned all of that out. And in looking at the slides this morning, I realized that they are not entirely vanity-free. You will notice that they are all blue. So we did a lot of that.

So this newsfeed rack. So I am going to get into some of the technical details because they are really necessary to understand what we propose to do in the future. So a rack is our unit of capacity. Now, when we say it is our unit of capacity, it means that all 40 servers inside that rack work together to produce the service that they are doing for the Company.

So a newsfeed rack has a lot of RAM and a lot of CPU. And the entire history of the last three days of activity in a very compact form fits on a single rack of servers at Facebook. It is just indexes into the recent activity. So when I am using Facebook and I go to Facebook and I pull up my newsfeed, what actually happens in the back end is a query goes from the Web servers to a newsfeed rack. And it has all of my friends, it has a list of all of my friends.

That aggregator which receives the query then contacts all of the leaves on all of the other servers and says what has happened with Jason's friends recently. It gets all that data together, ranks it via a ranking algorithm and then sends off the top 10 stories. So there is actually many more options. But it just ranks them and gives you the top 10.

Now, if you back up a little bit and think about the life of a hit on Facebook -- so a hit is a single request to Facebook -- what you have -- and I've drawn time going down -

- is a request starts and that Web server contacts mem cache. So it talks to our caching servers. It then authenticates the hit, makes sure I am who I say I am.

It might hit a newsfeed server. The newsfeed server returns a bunch of IDs. Those IDs correspond to stories, content that we want to show the user. All of that content is actually kept in the caching tier. So most of that is pulled up. Maybe some of it isn't there. We might hit a database, stuff some stuff into cache. And then grab some ads and then ship the page.

So this is a simplified version of what actually happens. In fact, we actually are shipping the page all along. So we actually ship the page in about eight stages. And that is so that the browser can start working on rendering in parallel with page generation.

So all of Facebook is made up of five standard servers. And so we have numbered them 1 through 5. And each one focuses on a single major service at Facebook. So the Web servers are all about lots of CPU. You need a lot of CPU to generate our website. The Database servers, it is everything to do with IOPS. So we use flash for that. Hadoop uses a lot of CPU and lots of drives. Then Photos, it is all about the lowest dollars per gig. So we store lots of photos. So we do a lot of optimization work there.

Now feed, that newsfeed service that I just talked about, uses both a lot of CPU and a lot of RAM. So we have copious amounts of both.

Any new project, any other service at Facebook that wants servers, they can have anything they want as long as it is one of these five servers. In fact, the capacity engineering team has T-shirts that say no on it. You really have to conform to one of these standard servers.

So what you get from there is you get volume pricing, which is huge when you are operating at scale. You also get repurposing. So if we have 10 services that all have a forecast and there is some uncertainty in the forecast, being able to take unused servers from one service and reallocate them to another is a huge efficiency win.

If we had a different kind of server for each service, then repurposing wouldn't be possible and we would always be dealing with the worse case of all forecasts. So it is a surprising win. But it is kind of analogous to, I guess, managing a mutual fund. Some things are up, some things are down. But across the board, you do very well.

You also get easier operations. So in other facilities, not our facilities, you might have a server-to-technician ratio of about 450 servers to 1 technician. Because all of our servers are the same, we are able to attain about a 20,000-to-1 server-to-technician ratio. So the servers are all very easy to maintain, very easy to operate.

Some of the drawbacks. So we do have five major servers -- major services. The other 35 services need to conform to that standard. So they might not fit quite perfectly. And there is also 200 minor services that all are important for Facebook. The other thing happens is the needs of the services change over time. And we will talk about how we have an idea to fix that.

So in terms of efficiency, our data centers are extremely efficient at managing heat. In a poorly-designed facility. for every watt of power that a server consumes, you are going to burn another watt just on air-conditioning. If you do some nice optimizations, you can get to about a 0.5 watt wastage. At Facebook, we have -- it is called a PUE of 1.07. I'm sorry -- we have a PUE of 1.07, which is the ratio of the total power consumed at the street divided by the total power consumed by all the servers. Which means that for every watt of power we consume, we are only burning another 7% on cooling. We do that by essentially eliminating all air-conditioning. We have talked about this project a lot. And it is really a huge efficiency win for us.

In terms of servers, the vanity-free design helps. What also helps is really running a good supply chain. So when you have lots of volume and you have very predictable purchases, you can really get a lot of cost out of those servers.

And probably some of the largest efficiency wins at Facebook come from our software. So we have a tremendous software efficiency effort. One that I would highlight would be HHVM or HPHP. We have talked about it a lot. But this is the core piece of software that replaces Apache and PHP in our infrastructure.

Now if we were to go back and use Apache PHP, which is very common, we would have to buy four times as many Web servers as we have today. And we buy more Web servers than anything. So this is a massive efficiency win for us. It is all open-source. We talk about it a lot. But software efficiency is huge for Facebook.

So in terms of next opportunities, I am going to talk about an idea called disaggregated rack. And then we will get into some new components that we think could be very interesting.

So when you think of that rack of newsfeed servers. And yes, there is 40 servers and they are all identical. But think about what they are actually putting online. You have about 80 processors of compute, about 6 terabytes of RAM, 80 terabytes of storage and up to 30 terabytes of flash. Now, the application lives on the rack of the equipment, not a single server; so we are not server-centric, we are rack-centric.

And so what we think we can do is we think we can take these components and we think we can stack them in a different way and put them online in that rack in a different way that can provide a really nice efficiency win.

So we think our building blocks of compute, which is really just a standard server; RAM. So a RAM sled. So a server with a lot of RAM on it; a storage sled, which is again just one of our NOC servers. So 15 drives. And you replace that fast expander

with a small server in the back; and then flash, where we have a flash appliance or a flash sled rather than individual PCIe flashcards.

And the three wins of disaggregated rack are that server/service fit. So when you have only five servers, you really want to be able to fit those servers -- -- you really want to be able to provide the exact right resource. And we will talk a little bit about that. There is the server/service fit over time. And then there is a longer useful life.

Now for that server/service fit, what you have on the left here is a Type 6 server. So a Type 6 server provides so much CPU and so much RAM. So the dotted line is the server. It provides this much RAM and this much CPU. The consumption is that arrow. And because we design the Type 6 server for multifeed of for newsfeed, it fits perfectly, the exact right amount of RAM and the exact right amount of CPU.

Now, if we give it to another service, say search, search doesn't need as much CPU, it is actually very RAM hungry. And so we actually only need about half as much CPU as what that service provides. Now if we differentiate the skew, if we give them a different type of server, then we start having problems with our volume, we start having problems with repurposing. And so effectively. that is a wasted CPU resource. It's not the worst thing in the world. But it is an area of waste.

Now the other thing that can happen is a server or service, their needs could change over time. So in the beginning, they might need -- they might be a perfect fit. But then a year or two in, they might need more RAM. So their CPU is fine but they need more RAM. Well all companies face this problem. If you have one thing that does the job and you run out of one bottleneck, you buy a whole other thing. So if I just bought 40 servers and then they need twice as much RAM, you are going to buy another 40 servers. Or if it is Facebook, you have 2000 servers and then they need twice as much RAM, you are buying another 2000 servers. So being able to do -- being able to grow RAM independently of CPU could be very important.

The other thing that you can get out of disaggregated rack is a longer useful life. So everybody everywhere gets rid of a computer when the thing that they need is no longer fitting. I don't have enough CPU, I don't have enough RAM, I don't have enough flash around a disk space. In general, most people refresh their servers every three years.

But with disaggregated rack, because it is just one resource, we think we can keep compute for more like three to six years. RAM doesn't go bad. It is just RAM; solid-state devices last forever. And so those could be five years or more, a disk sled easily four to five years and a flash sled could be easily six years or even 10. It's kind of crazy; flash can last for a while.

And so if you think of a disaggregated rack for graph search, rather than have 40 identical servers, we could have a mix of these building blocks. So, so many compute units, a couple, maybe one flash sled, a couple of RAM sleds and a storage sled. And the big thing that is different here is that later. So when graph search has a

CPU win, right they are more efficient in CPU. What that means is that they can handle more traffic with the same amount of flash and RAM.

Well if they are more efficient, then we want to add more flash or add more RAM and get more traffic going to that server. And the way to do that is to just add another flash sled. So in other words, if you look in the other direction, if at some point this particular service needs twice as much flash, it is far more cost-effective to just add in another sled of flash to double the amount of flash. So you have only paid for that incremental cost of flash, as opposed to buy a whole separate rack. So you are just adding that marginally-needed component.

The strength of disaggregated rack is we maintain volume pricing, serviceability, all of that. But then we are able to do the custom configuration. So better fit the services specifically. We can also do -- we can also add hardware easily over time and do smarter technology refreshes.

This also helps with just speed of innovation. So if you have only got to build a new component and it just has to support that new thing, whatever that new greatness is, you can just build that one thing and then slam it into a rack of all older or well-established SKUs.

Now, the potential issues there, physical changes are required. So those data center technicians are going to have to do a little bit more work. That is okay. We can hire more of those guys. And there is some interface overhead.

Now, the approximate win estimate -- so when I say OpEx, I really mean depreciation in power. Conservative assumptions show a 12% to 20% OpEx savings. And more aggressive assumptions between 14% and 30% OpEx. And these are just reasonable savings. So this kind of approach of disaggregating the components is something that of course we can do. But pretty much anybody in the industry can do. It just really requires being able to have your software adjust to this different model of compute.

So what are we really talking about here? Over the last 20 years -- so up in the upper left-hand corner is a large tower case. It holds a 386 computer from 1992. And down at the bottom is Amir Michael. And he is holding a Facebook server from 2012. Architecturally, nothing has changed between the 386 and what we are serving today. It is still a computer, a processor that assumes that all of the RAM is local, all of the drives are local, all of those peripherals are still on a PCIe bus. The RAM bus is all pretty much the same. Architecturally, the servers that we install in a data center are no different than a desktop from 20 years ago.

Now, there has been lots of innovations that have been huge. So the math coprocessor, that was great. It turns out if you have a dedicated chip to do math, it is much faster than a general processor.

SMP allowed two processors per server. So having two processors is better than having one because you can amortize the rest of the server costs over two processors, twice as much compute. Multicore processors, that was key in scaling Moore's Law. GPU's do vector math very well. And flash memory is the most recent game changer. Fundamentally, though, it is all the same thing; it is a processor running assuming everything is local.

And so while we have had exponential improvements in CPU, RAM, disk and NIC. And all of this has been amazing progress, we are still operating on the same model. And so one of the ideas, one of the things that we are hoping to accomplish with disaggregated rack and ideas like this is to break this model up, think about a server and a data center as the unit of compute is a rack. For any of you who have been doing this for a long time, this is no different than a mainframe. This is computers repeat themselves over and over and over again. All of technology goes through these cycles.

The big thing that has changed is the network bandwidth is now amazingly high. And whenever you have a lot of network bandwidth, whenever you have a really high backplane, you disaggregate your components. But this happens again and again. It is now time, we have 10-gigabits NIC's on all of our servers. It is completely conceivable that we can go to 25, 40 or even 100 gigabits in the next few years. So network is not a problem, which means that this kind of approach is perfectly reasonable.

Now talking about some of these components, we also see ways in which these components can evolve. Now with CPU, server CPUs are really just compute-dense versions of desktop processors. So the desktop world consumes lots and lots of processors. It might be a 4-core processor. If you double the cores, then you have got a server processor. Right? So you have got all of this nice volume in the desktop world and servers are just kind of hanging out on that.

One of the things that has been a very popular idea in the mobile world and other embedded processors is this idea of system-on-a-chip. So when you build servers and you build them over and over again, you notice that the CPU, the PCH, the NIC, they are all the same. So there is some components that you always buy together when you build a server. Fundamentally, these are all just standard IC chips.

So what system-on-a-chip is -- this is a popular idea that is gaining some traction -- is to when you are building a processor build the regular processor. But then leave a little bit more silicon, license NIC designs, license PCH. And essentially put all of that on to the same processor. If you are buying three components all the time, why not just buy one component? And so system-on-a-chip does that.

And the non-obvious win with system-on-a-chip is that it simplifies the rest of the computer. So when you are building a motherboard, when you are building pretty much any modern server, you might have an 8-layer board or even a 12-layer board that has a lot of complexity. Well if you pull in a lot of the components into a single

chip, then all of this becomes a much simpler design. And simpler means higher yields which means much cheaper. So pulling these things together, the stuff we buy anyway all the time, onto a system-on-a-chip design is a really nice one.

So really, SOC processors developed for servers that are derived from either the mobile or desktop markets can be very power and cost efficient. It is really a nice win, lots of people see it and we expect this to be very popular over the next few years.

In terms of RAM. So there is something called the DDR standard. So DDR standard talks about how RAM is designed. DDR allowed for the commoditization of RAM. So you have multiple vendors and they are all hitting a single standard. That DDR standard, it is difficult to hit; it is difficult to build a completely different kind of memory and hit the performance of standard RAM.

And so there is kind of no space for that in computers right now. You either have standard RAM. which is built on capacitors and consumes lots of power, or there is nothing. Then if you want something interesting, you have to put it off the PCIe bus, which is why you see PCIe flashcards.

And so we think that there is space and there is a lot of interesting opportunities for a separate class of RAM. And so what we expect is that for lower-latency memory technology, there is an idea of near-RAM, which is your standard RAM. And then far RAM. And where far RAM could be slower RAM.

It is like the stuff you are going to work on very frequently, the stuff you need all of the time, put that in regular RAM. The stuff that you need once every hour or once every 15 minutes, put that on the slower stuff. You spend more money on the fast stuff, you spend less money on the slower stuff. Everybody wins.

It's a really -- it's a nice analogy to how we have been using flash in our data centers. We use flash kind of as a RAM substitute. So we have a large data set that we need to keep. We can't keep all of it in RAM because RAM is really pretty expensive. Flash on a dollars-per-gig is both more dense, more power efficient and cheaper. And so we put on a lot of the data in flash and then keep only the more recently accessed stuff in RAM. That can be extended to regular computers.

So flash. So SSD drives are commonly used in databases and applications that need low-latency, high-throughput storage. And the flash industry has been focused on driving higher and higher write endurance and performance. So they have been going for better and better flashcards and better and better flash devices.

We actually think that by looking in the opposite direction towards very poor-quality flash, low write endurance, highly-dense flash, low IOPS performance -- so the opposite of what computers normally do is bigger, better, faster -- we want to go the other way -- we think that a cold flash storage option is possible.

So some of our workloads that we see and some of the workloads we think pretty much everyone will see is something where you want to write to it once. But then not read it very -- and you want to write to it once, read it all the time. But you don't have to update it.

So for us, that is photos. When somebody uploads a photo, we want to keep that photo forever. We don't want to change it; we are not doing editing on the photo. And even if we did, only a few people would do it. The fact is, is that a write-once, read memory flash for an alternative solid-state technology could provide extremely high-density storage at a reasonable cost.

And so just to run some very basic numbers on this, if you look at a rack of Knox drives -- and this is the system that we use for our cold storage -- archival storage -- you have about 2 terabytes of data in the rack. It draws about 1.5 kilowatts of power. And that is because many of the drives are spun down. So they are not consuming any power. It weighs about 2500 pounds and consumes about 0.8 watts per terabyte.

Now, if you compare that to a rack of SSD drives -- so we have done nothing special; just put a ton of laptop drives in a rack -- we didn't actually build it -- we did it on paper -- you get about 4 petabytes of data, it draws about 1.9 kilowatts of power and that is without any optimization. It weighs a little bit less and consumes about half as much watts per terabyte.

So the power density of a rack of drives in cold storage is almost 2 times that of a solid-state option that is built from currently available laptop drives. So no optimization yet; just buy stuff off the street. And we think that a focused effort in WORM solid-state options could yield much higher densities and longer hardware lifetime at a reasonable cost.

So with all of this massive data growth, today the industry produces 3 zettabytes of hard drives, going to about 20 zettabytes of hard drives in 2020. A large portion of that data that is going to be stored on those drives is really write once -- quite frankly, it is probably write once, read never. But let's imagine that we do read it. WORM could be just fine. And so we really think that solid-state for permanent storage could be very, very interesting.

And that is about it for me. Thank you.

Questions And Answers

A - Stephen Ju {BIO 6658298 <GO>}

Jason, we probably have time for one short question. But I mean you bring up the notion of the cold flash here. I mean, how is that materially different versus the other technologies that are already widely available?

A - Jason Taylor {BIO 18251157 <GO>}

Sure. So when we say -- so you asked how is cold flash materially different than current flash. So when everybody thinks about flash today, they think about NAND-based flash. And NAND-based flash is great. That is the latest incarnation of flash. In fact, we had EPROM and EEPROM's long before that. And those were designed with completely different silicon.

What we are saying here is if we design a solid-state device from the silicon up that is focused on very high-density data storage. But it is not mutable, it doesn't change, then we can get a much higher bit density at very low power.

And so it is an alternative approach to the whole silicon. But it is -- we have looked at it; it looks very possible and feels like we could probably get that to market in the next three to five years.

A - Stephen Ju {BIO 6658298 <GO>}

I think with that. we are actually out of time. So thanks very much, Jason.

A - Jason Taylor {BIO 18251157 <GO>}

Thank you.