Scribing Sessions DAY 1

Key-Value ----------

SILT: A Memory-Efficient, High-Performance Key-Value Store (Hyeontaek Lim)

Q1: Do you support update operations?
(Chanik Park, POSTECH, South Korea)
A: Essentially, updates are handled the same as inserts: add a new record as an insert. For update, we insert the data into the log. Then for read, we check the log hash table first and then the other stores.

Q2a: Given the properties of flash chips, which is that they only support a small number of overwrites in a single page per consecutive overwrites, how do you deal with very small entry sizes? You talk about Twitter-sized entries, which can be as small as 32 bytes. Is there a practical limit on how small entries can be?
(Ted Wobber, Microsoft Research Silicon Valley, United States)
A: There is no limit for small entries. In fact small entries are always good and (?). Because we already (?) any writes to a sequential write. If we have very small writes on inserts, insert converts that to block writes, because we haven't seen no such performance for...

Q2b: I see. So you don't make it stable then right away? Per 32-bytes write, you are going to have a hundred of writes per flash page. You can't do a hundred of writes on a single flash page, so the FTL on the SSDs will have to move it to another page, isn't it?
A: No, if the (?) size is very slow, you can do a full-size (?) because if we flush at high rate. This rate would still be low, so that you can be sure you don't flush bytes. If it becomes very high, we can put the bytes together and revert to block write.

Q3: I didn't see any performance numbers for writes in the paper. Would be dependent on the relative sizes of the 3 stores? Would there be a bottleneck in merges?
(Marc Shaprio, INRIA & LIP6, France)
A: Yes, we need to frequently merge the log store into the stored store. We ensure that the size per log store is large enough. We don't want to encode too many flash writes so we always have enough size in the log store. I haven't shown it, but we think where is an intermediate stage between log store and stored store. The hash store can store 15-20% of the data items, so we don't need to perform merging frequently.

Q5: I have a comment: Since your storage system is log based or indexed, you don't really need the second logging inside the flash (FTL), but this would make a very good storage system for flash on its own.
(Burton Smith, Microsoft Research, United States)
A: The flash performance is not as good for random writes as for sequential ones. We are able to run our system on any storage device.

Q6: During a read, if there is a hash collision, maybe you can somehow check this later. But during writes, if there is a hash collision, although the probability is low, it seems that you might overwrite some existing value.
(Yubin Xia, Parallel Processing Institute, China)
A: Even if we only use a single hash, we expect that a 160 bit hash keys is sufficient to avoid collisions. The probability of collision should be very low.

Scalable Consistency in Scatter (Lisa Glendenning)

Q1: First a comment: scalable consistency sounds misleading, because consistency is a yes or no question, whereas scalability is not. Then the question: What is new?
(Rachid Guerraoui, EPFL, Switzerland)
A: To the best of our knowledge, we are the first to have strong consistency across a wide area network.

Q2a: There are obviously issues with strong consistency liveness assumptions about if you have various partitions in your system or failures. The question is: you were targeting a relatively small number of... Do you know of any measurement studies of uptime known behavior in vuze that indicate whether or not you would be able to provide these guarantees under the churn you actually see in these systems?
(Mike Freedman, Princeton, United States)
A: The churn we used in our experiments is actually based on measurements that use peer-to-peer systems, using the Pareto distribution. The lowest churn rate that we evaluated is 100, which has the mean life-time of 100 seconds, which is actually pretty impressive. It is possible that we could test our system in an even smaller churn rate...

Q2b: I guess the question is not if you want to say you achieved strong consistency, but what is the churn for the average group that actually if you look at the tail of the distribution other groups for which you had too many faults and therefore in some sense you can't achieve linearizability and therefore the question is what do you say, you obviously need to give up liveness?
A: So you're asking is what if you have group failure. In this case we always achieve linearizable consistency, but yes we will lose liveness in that case.

Q2c: So you achieve safety, but lose liveness...
A: Yes.

Q3: What happens when you lost more than *f* members in a group?
(Hakim Weatherspoon, Cornell University, United States)
A: So, what happens if a group fails? We designed the system so that there is a very low probability for that, but cannot be ruled out this comes down to unavailability in key space. For future work, we think that groups can take over key space if adjacent groups fail.

Q4: A couple of slides back, you showed the cost of consistency by comparing it to OpenDHT. The implementation of scatter uses policies and optimizations do load balancing. If we included such policies in OpenDHT, would the difference in this cost increase or still be

comparable?
(Hussam Abu-Libdeh - Cornell University, United States)
A: OpenDHT has been through a lot of iterations and they have done lots of optimizations already. Both OpenDHT and Scatter could be further optimized and both results would shift. It's hard to say.

Q5: I was wondering whether in your design you think of the group, the processes or the machines in a group, being close to each other or do you allow the machines to be in geographically distant locations?
(Marcos Aguilera, Microsoft Research Silicon Valley, United States)
A: The question you are asking is what about correlated failures. If you are optimizing groups such as all of them are close to each other geographically to save latency than you can have some correlated failures and never hide a failure, or have a higher probability of failure. So the answer is we have actually a number of policies, which we did not talk about in the talk, but are in our paper. We actually have a number of policies that we implement in Scatter, one of which say we evaluate the overall pattern of a group fail, so in that case we optimize for the large group sizes. We also have a separate policy saying we really value robustness, and in that case Scatter groups would address themselves that they are over a wide geographic area, of course in that case you see more latency.

Q6: I know if you are dealing with faults that are a little worse, but more malicious, you probably can't do some of the optimizations you do, like divide the keyspace among primaries, but... My bigger question was: if you have within each group, if you are running a more end-to-end protocol, something that requires all-to-all communication, do you have any idea how it would affect the performance in your p2p setting?
(Siddharta Sen, Princeton, United States)
A: My intuition is that performance would get worse. I don't have a more concrete intuition on that. Again, we have optimized the system a lot to get the performance you see, and I suspect a lot of the optimizations would so be impossible or some things you can do with a different set of protocols. Perhaps we want to address Byzantine faults and other faults that we don't have in our current system.

Fast Crash Recovery in RAM Cloud (Diego Ongaro)

Q1: One of the issues is to improve scalability for reads. What happens to read hotspots?
(Khaled Elmeleegy, Yahoo! Research, United States)
A: We believe that that we don't need replication for performance. We might be wrong though. Our servers can issue a million requests per second. We don't think most applications need more than that. We're fast enough.

Q2: It seems that your main technique that you are using just to scatter the backup actually creates a trade-off which is that if you need to recover lots of machines, or say all of the machines, when you have a power failure for example, you have to pay extra overhead because you have all of these messages flying left and right on the network to recover all of those, and I was wondering if you have additional techniques to recover fast in that case, for example perhaps writing to the local disk in each machine in addition to...
(Marcos Aguilera, Microsoft Research, United States)
A: That's a great point. So, what would you do for cold start, what happens when all the data is gone. So, we're actually kind of in the middle of designing and implementing that (…). But I think that at the end of the day what you would go to end up saving is basically network bandwidth during network recovery time. At the end of the day we have plenty of network bandwidth, so it's ok to basically kind of unscatter all the stuff and restore onto individual hosts, the network is not going to be the thing slowing you down there. It's just going to simply be the fact that you only have so much disk bandwidth which to bring all these hosts back. So we could do something like store things locally, the problem with storing data on local disks is then you are kind of limiting yourself to the disk bandwidth for writes on that particular host, we buffer these updates. And it turns out that when we scatter we can actually buffer data into memory of one set of backups, and then the next set of backups, as you keep it ahead of one, switches have burst bandwidth that can exceed the speed of disks for a very long period of times in Ramcloud. We want to keep up with those properties.

Q3: When writing, you mentioned synchronous writes, why not use asynchronous writes?
(Joel Nider, Qualcomm, Israel)
A: We are delaying synchronous writes, doing it background, buffering updates in RAM until a more convenient time. So it is asynchronous. We consider an object durable after it is stored in enough other machines' buffers.

Q4: If you do [asynchronous writes], what happens during a power failure? What are the implications for data loss?
(Marvin Theimer, Amazon.com, United States)
A: That's a problem. Other systems have this problem too, e.g. BigTable. BigTable will write data to GFS, which pushes it to the OS's buffer cache,

which you hope the OS will push it out before a power failure. I'd count on battery-based backups in the data centre. There are also NV RAM controllers, and flash-backed RAM with supercaps – using these, when power goes out, it quickly copies data out to flash whilst running on the super cap. There are some things you can do there but we're not better off than any other system.

Q5: This is great work, by the way. One question I have is: a couple of times you mentioned both low-latency data center interconnect, but also a data center interconnect that is homogeneous in terms of the bandwidth it delivers. How dependent is your system on that homogeneity assumption both in normal time operation and also during recovery?
(George Porter, University of San Diego, United States)
A: That's a great question. We basically designed the system from the ground up relying fully on the fact that we would have full bisectional bandwidth in future datacenter networks. That's probably a reasonably controversial thing, so we are kind of counting on it. I think that are a lot of things we can do. I think during normal operation there are not any major problems. It's generally during recovery when there is this huge cross-section of bandwidth using. In the end, what it would probably mean is that if you didn't have full bisectional bandwidth, is that you want to come up with some form of locality for backups. You want to scatter your data broadly enough so that there are plenty of backups always scattered, but at the same time you would to have to localize so that you are not going to cross whatever switch it is you are trying to avoid, that's a bottleneck. We have talked about this a bit here and there in our design meetings. I think it is fully possible to do but until we have a need for it, we wouldn't worry about it. Right now, we are able to achieve full bisectional bandwidth using our small two racks so...

Q6: In the paper you talk a lot about the recovery time. In general use, would write throughput be limited by the bandwidth of the network? Or, perhaps, by the processing time of the back up server? What about throughput under normal execution? How do you compare with memcached? Or would the router in the switch be the bottleneck in normal execution?
(Jin Chen, Fudan University, China)
A: We are limited by disk performance eventually. If a single host is seeing high burst of bandwidth, you can buffer a segment on many backups. We take 2 round trip times for each write request. Right now we're seeing something like 14 microseconds round-trip for 100 write requests. It's a good question, you're network bandwidth is down by one third because you're tripilicating objects. But with 10GB ethernet, that's still several hundred megabytes per second. The disk bandwidth is really is the more limiting factor.

Q7: I think you guys did a really cool job in the particular way the disk bottleneck and really leveraging the size of your cluster, but if, assuming that the network bandwidth is not the bottleneck in your system, does that mean that, in its current state, the performance of

recovery will be proportional to how good your NICs are. Let's say that if you didn't have the Infiniband NICs you are using now, you just have regular...
(Siddharta Sen, Princeton, United States)
A: OK, so that's a great question actually. In particular, we get 25 Gbits/s and not 10 Gbits/s, so that's quite a bump. So the answer to that is it's not really a big deal, it just means that you have to scatter your recovery a little more broadly. So, it turns out that with 10 Gbit, if you would have a 10 Gbit network, you would have to recover something more like 300MB partitions instead of 600MB partitions. Particularly the problem is basically re-replicating the log contents during recovery causes kind of a three-fold explosion of data coming out of the recovering the master, so...

Q8a: When you have a failure, you're proposing to take the capacity of the data center and focus it at the job of restoring the failure. From a usability perspective you have a pause in availability. When you scale the data center, you're going to have more failures and more bottlenecks. Your recovery times will increase as data centers grow larger. A data centre with lots of faults will be on and off with periodicity of the faults.
(Alex Snoeren, UC San Diego, United States)
A: True, we could limit recovery and try to proceed with servicing requests in the mean time. If your application is very data intensive, you're machines are after the data. So any time you're taking pauses to recover, it's better to get that data back online and start processing again.

Q8b: Do you understand how multiple simultaneous failures scale? E.g. recovery of 2 servers instead of 1.
A: We haven't done multiple simultaneous failures yet. It will probably be one of the next things to work on in the next few years. At the end of the day, we can recover from multiple simultaneous failures much the same way, but it's very resource intensive. The question is how to balance it. It's a tricky issue.

STORAGE ----------

Design Implications for enterprise storage systems via multi-dimensional trace analysis

Q1: So k-means clustering assumes that dimensions are of comparable quantity. Did you think about other methods like PCA, which seems more natural?
(Sue Bok Moon, KAIST, South Korea)
A: So the question is, why k-means why not say PCA that is more natural. The reason is k-means does not make any assumptions about the nature of this multi-dimensional space. In our method, we do some localization, so we normalize all of the dimensions to [0,1] range. We actually considered PCA but the reason we ended up not using PCA is specifically because within PCA there is an assumption about multidimensional linearity. K-means doesn't make that assumption. PCA gives you a vector as a principle component, which will be 0.5 of the IO size, 0.3 of the sequentiality and 0.78 of the overwrite. So, PCA is helpful but it doesn't lead to the same answers.

Q2: How your design of the application would apply to a different enterprise network?
(Petros Maniatis, Intel Labs, United States)
A: That's a generalization question, we do not know, I encourage applying the same methodology, which is general: the implementation must be tuned to the particular use case.

Q3: It seems to me that you make this assumption that if you see some pattern in your data and you apply some optimization based on that, you will see a performance improvement. Whereas, to give an example if you look at client consolidation and you said just by looking at read/write ratio you can get better consolidation. Another factor that may be critical is how are they correlated in terms of their uploading time. If clients are sending the workload together that is differently. Did you implement most of these techniques or some of these techniques to see that what you learn really leads to a performance improvement?
(Ajay Gulati, VMWARE inc. United States)
A: That is a fantastic question. Thank you! So, time-series analysis at a fine-grain level we have not looked at that just yet. I imagine, you guys at VMware will have a lot of insight about that, and you would like to increase the way you describe an access pattern, even further. So, you look at that from storage, you might also want to look at that from CPU, memory, etc.
So, that analysis will lead us to peaks and valleys, but I think the way I will interpret that amount of data is that if you have peaks in the arrival patterns, that is really initiated on the customers, therefore is out of control of the system. Being still, the responsibility of the virtualization software module to service that. And the insight is to understand how they arrive, what kind of

peaks, whether is just sequential and so forth. That will be one away of doing a fine-grain time-series analysis.

Differentiated Storage Services

Q1a: How can you get to the metadata, because from the filesystem perspective that is the part that includes the information about the file?
(Yubin Xia, Parallel Processing Institute, China)
A: The filesystem knows the structure of the files, not the contents. The corresponding application keeps the latter information. Once you go to the application level, you can get more information about the content of the file and thus you can provide more classification details. Apart from only data vs. metadata classification, you can have other classifications.

Q1b: So you have to trust the application to provide the information exactly.
A: Yes, but the application is the database and there is a handful of databases that matter, and a handful of filesystems that matter and these guys can play well together in the way we think is feasible.

Q2: I have a question about the coalescing you did. You said you only coalesce same class requests. Now, have you looked at the impact of that because you could imagine that having a positive as well as potentially a negative impact, might reduce the overall amount of coalescing you do, or might give you better coalescing. Have you done any evaluation of that?
(Malte Schwarzkopf, University of Cambridge, United Kingdom)
A: It definitely has a negative impact if you do not do anything with the uncoalesce request. So, if you just do not coalesce then you are actually sending more I/O. But, what we have shown with SPECsfs and with the database workload, is that even when you do not coalesce as aggressively, you make up for it in the backend when you get the small and important things cached. The trick is you have to pay the tax of not coalescing, in order to get the benefit of what you can do in the backend storage system.

Q3: So, a little earlier we were saying that people would describe what the data is. And this last slide you were describing how people instruct the system what to do. I can imagine lots of people will ultimately decide on a special flag to the storage system to tell it what to do. There is a reason you chose to go with the description of what it is instead of what to do?
(Daniel Peek, Facebook, United States)
A: The reason that we separate classification from policy is to make life easier on the person developing the software. As a result, you classify your data once and later on depending on the storage system, you will find that happening. So, in our EXT3 prototype, if we would have gone in there and just hardcoded priority levels, it would make sense at the time, but if a new storage system comes along and it is not based on priorities, we would need to go back and think about what each of those priorities meant. Setting priorities from the beginning "would declassify stuff one", it is a level of

indirection before you do the actual policies on them. Therefore, it is easier from the software perspective.

A file is not a file: understanding the I/O behavior of Apple desktop applications

Q1: Have you looked at other type of files? Netcdf and HDFI?
(Swapnil Patil, CMU, United States)
A: No, we have not looked at other file types.

Q2: I wonder if you could comment on where filesystems went wrong.
(Peter Chen, University of Michigan, United States)
A: There are different approaches on file systems, so it is hard to make a general statement. However, some issues to think about are atomicity, selection of the right API and support for different workloads. But it is hard to say.

Q3: I guess the obvious question is: how much is this specific to Apple? And, is it just bad programming?
(Marc Shapiro, INRIA & LIP6, France)
A: It is hard to say how much it is specific to Apple. It would be great to do a parallel study, maybe with Microsoft Office or other classes of applications. Whether that is bad programming, I mean you always want to be efficient in programming and it is hard to understand what your frameworks do. So, maybe there should be better tools so you can understand that your framework has these applications. It is not necessarily an important thing from the developers' perspective.

Q4: Do you think your conclusion "File is not a file" applies to server programs, for example database programs that use files as file streams.
(Yufei Chen, Fudan University, China)
A: "File is not a file" states that there are a lot of file types (I'm not very familiar with a lot of file programs but there are a lot of other file programs out there) that show a more complex structure.

Q5: Does it make sense to change how the files are laid out depending on their usage behavior?
(Davide Frey, INRIA Rennes, France)
A: Different applications have different behaviors and to be able to reach a decision we should actually look at a bigger pool of application but I hope this gives an idea.

Q6a: How much does this actually matter? Is it really a big problem? What would be the benefit of doing it all right?
(Costin Raiciu, Universitatea Politechnica Bucuresti, Romania)
A: We have been looking at the while the applications behave sluggish, and we suspect that much of this was due to I/O.

Q6b: If we go to cloud platform, then it might matter even more?
A: Yes, certainly.

SECURITY ----------

CryptDB: Protecting Confidentiality with Encrypted Query Processing

Q1: I understand you compute these columns as and when needed; you peel off the layers of the onion as necessary. Have you considered the execution over time, for example running for a year, you would start to see extra columns; have you considered something like a cleanup procedure to remove excess columns?
(Malte Schwarzkopf, Cambridge, United Kingdom)
A: For many applications the query set is fixed, the same set of queries will be issued over and over with different constants. For those applications the layers of the onion will stay the same. For the applications we looked at most levels of the onion stayed at RND. For applications such as analytics, a technique such as you proposed would be useful.

Q2a: So, in case of user-based keys, how do you perform the onion stripping for aggregating results from multiple users? Because you would need keys from all users to perform queries…
(Volodymyr Kuznetsov EPFL, Switzerland)
A: If you are talking about operations over data belonging to multiple users, two issues have to be considered here. First, is the data decriptable? Is the current user logged in? If the user is not logged, we cannot decrypt the data. If user is online and wants to perform the query and aggregate data, we can have a proxy to aggregate directly the data. There's even a better possibility, programmer can provide more clever annotation.

Q2b: But how is strip on onion layers performed?
A: In the multiuser case, you only strip off the layer for specific users. Moreover, ahead of time you can start with the correct onion layer.

Q3a: I really like the work. I wanted to ask if in many databases there is a lot of sharing of almost all the data between many users. In your second conclusion you talk about what happens when users are logged out, but I guess that means what happens when all users are logged out because of the key chaining.
(Brad Karp, University College London, United Kingdom)
A: Right, so basically a piece of data that belongs exclusively to logged out users won't be accessible.

Q3b: What happens when most of the data belongs to some logged in user? Do you have any thoughts about what else you might do in that situation?
A: Right, so you might have a person such as the administrator who has access to a lot of data, when that administrator is online, the attacker

attacking then could get control of a lot of data. One measure of precaution would be for a person with such high power to have multiple roles, one of which is for a normal user and sometimes, hopefully for a shorter period of time and more rarely use the full administrator privilege. It is part of our future work to reduce the amount of data of logged in users that leaks.

Q3c: I think there exists a case where it's not just an administrator and users share all the data. I think that case exists too.
A: That is part of our future work to try to do that.

Q4: So, you use encryption schemes, which selectively leak stuff, order, whatever. If I was an attacker, I might try a variation of a known plaintext attack such as a "chosen query" or chosen plaintext attack; have you considered these kinds of attacks?
(Paulo Veríssimo, University of Lisbon)
A: When I specified the first threat, we considered a passive threat; where the adversary could only look at the data but not issue queries.
In the second case, where we consider active attacks, we don't guarantee the protection for those logged in users. So, what you are saying is already included in that case.

Intrusion Recovery for Database-backed Web Applications

Q1: What if the user inputs it depends on previous output of the web? If the output is wrong the user input may be wrong consequently.
(Yubin Xia, Parallel Processing Institute, China)
A: For the applications that we saw, our DOM replay works ok. For other applications, the user input may depend on the output. In that case, we could allow the application to specify to WARP that certain bits of input may depend on output, and act accordingly.

Q2a: I think this is actually a follow-up question to that one, but it should be obvious to any Star Trek fan: What if you have external dependencies that are external to the actual timeline? Say I'm using this photo sharing site and Alice, who I respect a lot, is a victim to one of these attacks, which causes her to share all her photos with me. I look at it and say, "Well if she did it, I should probably share mine with her." But, I would normally not do that. And now, during the replay there is this external dependency. What are your thoughts on that? I know it's a hard problem, but you've thought about it a lot, so...
(George Candea, EPFL, Switzerland)
A: The problem is there because the system doesn't have a way of tracking the external dependency.

Q2b: It's external...
A: It's in the user's mind. The user did something based on what he saw someone else do. In these cases, one thing that you could do is, once a repair happens, the user can go back and look at what repair happened and whether that repair would change any of the stuff that he did. It's a way of auditing what the repair actually fixed. Potentially, the user could then reapply or undo his action based on this action because of the semantics.

Q2c: You would basically announce to the users...
A: Yes, but it is difficult to separate events that are legitimate of the events that compose the attack in this case.

Q3a: My question is how to you compare DOM nodes at replay time compared to logging time, essentially how do you ensure you have the same nodes?
(James Mickens, Microsoft Research, United States)
A: What we do is searching for the same nodes based on id based on path and we check the type of DOM elements to ensure that it is a text box and matches.

Q3b: The reason I asked that question is that there is a dependency on how you do that equivalence mapping. There are some things that are

not enumerable so there are ways an attacker might hide elements that are not enumerable unless you are checking all properties.
***[Clarifying question]* Are there any things that aren't enumerable in the DOM? If you're not comparing all properties is there a way an attacker could hide elements?**
A: Since we do it in an extension we can look at all elements of a DOM entry.

Q3c: Do you do that in practice?
A: At the moment we check a subset of properties, but that's an implementation detail.

Q4: This is an extension to the previous question - is there any attack vector opened up by the fact you have a browser extension uploading data - can I modify that extension in some way; is there something i can upload that would allow compromise?
(Malte Schwarzkopf, Cambridge, United Kingdom)
A: A very good question - we are careful when we run the shadow browser on the server side - we ensure the user can't do anything he wouldn't be able to do on the client side; there is the possibility the user would be able to undo certain things; in the current design we only allow the administrator to undo things to avoid exactly what you describe.

Q5: How do you protect the Warp data at the server; what if we attack that?
(Eric Jul, Bell Labs, Ireland)
A: Warp data can be made read-only or it can be sent to a separate server so we can protect it that way.

Software fault isolation with API integrity and multi-principal modules

Q1: What you are fundamentally trying to do is preventing a kernel module from misusing another one? Wouldn't it make much more sense to put all modules is separate user processes and run them in userspace? It seems the right way to do it.
(Andrew Tanenbaum, Vrije Universiteit, Netherlands)
A: I agree.

Q2a: Thanks for doing this follow-up work for XFI - XFI is very cool and useful and not very well understood. However, I felt your example of the spinlock was kind of bogus. For example, why would I pull a spinlock outside the module when I could just hang the kernel by holding the lock anyway? Wouldn't it make more sense to just inline the spinlock call?
(Jacob Hansen, Bromium Inc, Denmark)
A: Yeah it's true that is a bit of a toy example; but there are more reasonable devices such as a PCI card where the interface is more complicated.

Q2b: I think what works for Microsoft is that they have a very nice kernel API and that's what we need for Linux.

Q3a: So, I understand the basic notion of what it is that you're trying to do, but a question that comes up to me is: "Ok, you assume originally that the kernel API was correct, it wasn't, and bad things happened." Now you have to assume, by faith, that the LXFI annotations are correct, and if they are wrong, you're doomed. Where does the madness end? Where can I stop assuming? Is that the smallest thing I have to assume?
(*Petros Maniatis, Intel Labs, United States*)
A: One thing about this [unknown] is that, suppose that, for the function, there is the probe function pointer of the PCI network driver. The PCI network driver may create an instance of such a function pointer because it will expose the callback to the core kernel. The contract on the probe function has some implicit rules, which are assumed by the core kernel when it calls the probe function, and then you can express all the rules on the probe function pointer, which will force all these implementations to follow.

Q3b: I understand the mechanism, and it makes a lot of sense. I as user of the system am trying to get some guarantees out of it. I might want to get guarantees until the cows come home, but if my annotations are incorrect, I'm going to get guarantees about meaningless annotations. So the question is: How do I know (I or my customers) that

the guarantees are meaningful? How can I connect the annotations to something that says, "No privilege escalation". That is what I am asking: is this helping me get to that point, or do I have to do a layer of annotation on the annotations, to make sure that these are correct?
A: I don't know the answer.

Q4: I was wondering if you could infer some of your annotation using a static analyzer. It seems to me that it could also be used to verify the annotations?
(Gilles Muller, INRIA/LIP6, France)
A: Yes, we thought about that. The problem is to infer members of structures from pointers tag. Maybe one could add annotations to check the type of arguments, but it seems it would become harder to use.

Q5: Quick question: you've quantified the CPU overhead for all of the extra code that's been introduced, but have you quantified all of the memory overhead, how much all of these extra structures need space-wise?
(Bernard Blackham, NICTA/UNSW, Australia)
A: No.

Q6a: You've introduced these annotations, but it seems, like, to use them properly, you really need a different way of thinking about your code.
(Ivan Beschastnikh, University of Washington, United States)
A: Yes

Q6b: And, of course, you've shown that perhaps I don't need that many, but you've written them. What makes you think kernel developers would want to adopt this methodology?
A: Well, could you repeat that again the last sentence?

Q6c: It seems like there is a software engineering methodology that is implied by these annotations. So to use them properly you kind of need to reason differently about your code. So what makes you think that kernel developers would want to learn this approach? Kind of beyond the API, there are other things you need to think about?
A: I don't know if Linux developers would think it is a good idea to involve the API security in such a way.

Q6d: So is it easy to write these annotations? Are they easy to learn?
A: I am not an expert in Linux kernel developing so I'm not sure, but maybe.

REALITY ----------

Thialfi: a Client Notification Service for Internet-Scale Applications

Q1a: You might talk about this in the paper, but in your design it seems like notifications are cached in a row in big table. I was wondering if you have thoughts about how this would work if you have a large number of clients and you want to, for example, do a software upgrade.
(Michael Freedman, Princeton, United States)
A: You're talking about if you have a huge fan out. This is an actually an issue that has come up many, many times. And the way we sort of work is that as applications ask for features, we think about them and then we provide them, and this feature has come up but not really. So the thing that you have to do for this is you have to make sure that when you do these propagations, from the master to the registrar, is right now for example internally what we end up doing, is that we propagate, and if it fails, we retry the whole thing. You can't do things like that; you have to sort of do partial propagations. That might be obvious, but the other thing one has to realize is that the amount of state we are maintaining in memory is actually proportional to the amount of online clients. It is not proportional to all that clients that have registered for it. Depending on the application you're looking at, in many cases (not all) the typical number that are seen across a variety of applications across the last few years is 10% of your clients are online at a time. So, we are not maintaining state and memory proportional to all the clients.

Q1b: Yeah I mean I wonder if you know, Justin Bieber has 5 million followers. 10% of 5 million is 500,000 followers all in one row in Bigtable, is a lot.
A: Sure. You could imagine starting to look into alternate ways of doing this. In fact, at that point you may even consider a hybrid solution of doing polling. I mean, this system is designed for, what you care about is getting things done really quickly. If you get somebody's tweet in 3 minutes, is that the end of the world? I don't know.

Q2: When a client goes down, you need to recovery from partitions, where do you recover from: the local partitions or essentially from the servers?
(Marvin Theimer, Amazon.com, United States)
A: Yes, when one client goes down, it needs to recovery its status. Whether from the local partition or remote server depends on whether the local partition has the client status. If it doesn't, it needs to get the status from the server. What we do is to use a best effort protocol to poll the information required for recovery from the server.

Windows Azure Storage: A Highly Available Cloud Storage Services with Strong Consistency

Q1a: You didn't talk about what happens during primary failures. How do you handle those in your system? Since you said the role of the primary can't be reassigned.
(Ivan Beschastnikh, University of Washington, United States)
A: When a primary fails, the partition layer of the client will realize it very quickly and create a new extent and continue.

Q1b: And what happens if it's partitioned, and it comes back online?
A: The same situation handler which I showed for the other node in the examples, where you'll come back online but it won't allow taking any appends until it talks to the master.

An Empirical Study on Configuration Errors in Commercial and Open Source Systems

Q1: In your presentation, you suggested provide as few knobs as possible. On the other hand, application may require more parameters to achieve flexibility. In order to be flexible, there is some conflict between easy to configure and flexibility. In this case, what's your suggestion?
(Yufei Chen, Fudan University, China)
A: From our point of view, it's more like art of design instead of science. There are indeed some parameters that need to be set by users like network settings. However, generally we suggest if the parameter can be automatically configured, you should automatically configure it.

Q2a: So the contribution of the work is a sort of practices that developers should carry out in order to avoid configuration errors. Software engineering suggests another best practice, which is configuration testing. In fact, there are tools to explore the space configurations automatically. Do any of the projects that you studied perform configuration testing, and if they do not, how do you think the results would change if they did?
(Ivan Beschastnikh, University of Washington, United States)
A: To the best of my knowledge, I don't know if any of the systems we studied applied the technologies you mentioned.

Q2b: Do they test their configurations in any other way?
A: So, I don't know if they test the configuration, but some of them provide configuration tools to help them configure or help them decide on the configurations to use. I don't know if they apply the configuration testing process.

Q3a: If you were to weight bugs instead of by bug report, instead by the pain and frustration that they cause, do you think that would change your characterization, and if so, how?
(Jason Flinn, University of Michigan, United States)
A: I think it's probably hard to measure the frustration in a quantitative way. What we did is that one of the aspects we studied is the diagnosis time: basically we measure from when the person first asks the question until it works. It's kind of an approximation for frustration.

Q3b: And?
A: And we have the results in the paper. There are many good results in the paper that I don't have time to cover.

Q4a: How do you know whether ambiguous error messages are the cause of taking a long time to resolve the problem, or some kinds of failures both have ambiguous messages and can be hard to diagnose? ***(Ariel Rabkin, UC Berkeley, United States)***

A: As I said before, it's very hard to understand configuration errors because it's just a configuration. We tried our best to get all the information we can get from the forums. If some message does not actually help to diagnose, we call it ambiguous.

Q4b: That's not what I asked. How do you know that the ambiguous error message made the thing take a long time, rather than just some other common factor?

A: If by saying this case had an ambiguous message, we mean that the message didn't really help to diagnose the problem. Then we used the approach mentioned to measure the diagnosis time. There might sometimes be cases when there is other affects on the diagnosis time, especially for open source because maybe some expert takes a vacation and they are not looking at the forum. To answer your question, we don't know exactly if it's caused by an ambiguous message, or by some other thing.

Scribing Sessions DAY 2

VIRTUALIZATION ----------

Cells: A Virtual Mobile Smartphone Architecture

Q1a: I love this work, congratulations it's wonderful work! I did have a question: in general, people do only one thing on the screen at a time today, but is this a chicken vs. egg question, where if they had the ability to have multiple things on the screen at the same time they would use it? I actually do run multiple things at the same time.
(Mary Baker, HP Labs, United States)
A: So you actually run multiple things at a time? Since the screen is so small, how can you do that? Generally people only run one application.

Q1b: Right, but is going into the future will this be a reasonable assumption?
A: We have thought a little about this and how to extend this to a tablet, and we thought of interesting ways of splitting the screen, splitting the device resources, this is an area we are looking into.

Q1c: OK, I'd love to talk to you some more about it.

Q2: How do you deal with notifications? If one virtual phone has the entire frame buffer and the other phone gets a text message and wants to pop up on the screen to tell you while playing Angry Birds "oh you have an important text message".
(Craig Souls, HP Labs, United States)
A: This is a little bit of future work. We didn't implement things, but the basic idea is that we have a daemon process running that can proxy these things between virtual phones and so depending on how you might configure your device maybe you don't want text message notification showing up in your work phone from your own phone. So you have to configure this but you can use the daemon process to proxy those messages back and forth.

Q3: Why don't you [.....]
(Yufei Chen, Fudan, China)
A: Yes that is a valid question, the answer is to have accomplish isolation, have specific policies, multiple user accounts, you can enforce the security by that

Q4: First, I'd like to see this is a very nice phone with a really powerful processor (audience laughter). You mentioned that you can actually have multiple incoming phone numbers, how does it work given that the phone only has a single phone number?

(Joel Nider, Qualcomm, Israel)
A: We also ran this on a Nexus S and on a NVIDIA tablet. In response to your question about multiplexing, we use a VoIP service, with the number registered on VoIP. CellD intercepts incoming caller IDs, attaches a digit to the end of the number, and thus can channel it up to the appropriate VP.

Breaking Up is Hard to Do: Security and Functionality in a Commodity Hypervisor

Q1: So talking about isolated every device is more complex can, the thing share complex, why is that? No bugs that left shared
(Patrick Wendell, UC Berkeley, United States)
A: we assume that there is bugs, but it is [...]

Q2: Why the performance of wget, shown in the graph, was better?
(Benard Balckham, UNSW, Australia)
A: Standard xen installation, there is more interaction of those 2 components, isolated the cpus we do not how much weight should we put on each component. [...]

Q3a: To see if I understand the sharing model, do I have to have to isolate multiple devices?
(Alec Wolman, Microsoft Research, United States)
A: Yes, or we can virtualize.

Q3b: As rolling-back devices, do you have one to reset hardware and that became an issue…
A: Yes, if we just rollback software and hardware, yes it becomes an issue on terms of performance.

CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization

Q1a: What are your plans with regards to memory overcommitment, because there appears to be a problem with swapping guests out or deduplicating guest memory with this approach?
(Orna Agmon Ben-Yehuda, Technion, Israel)
A: Are you asking what happens if the memory is swapped to disk? The hypervisor will actually access the virtual machine memory but CloudVisor will assume that it is not legal, so it will encrypt the memory contents before giving the memory to the hypervisor, so all the contents will be encrypted when the hypervisor writes it to disk; When the hypervisor will read it to page it back in, CloudVisor will decrypt it back.

Q1b: OK, but what about memory deduplication -- where two VMS have the same page in memory and the hypervisor only keeps one copy in physical memory.
A: That will be a problem here and we currently do not support it. Maybe that will be future work?

Q2a: How do you protect the disk or other device? For example running a legacy OS in VM, the real time clock can be compromised, how do you know that you will not expose the security?
(Stefan Saroiu, Microsoft Research, United States)
A: How to handle the IO besides disk? In the design of Cloudvisor, we identified two devices in network and disk, kept integrity and privacy; network should be all the traffic.

Q2b: But I mean how do you do secure clock?
A: Yeah, its a kind of attack if the hypervisor tries to manipulate of the clock vm, maybe will cause a random behavior (…)

Q2c: In summary, you say that if I can compromise the clock, I cannot compromise the confidentiality of the machine
A: I think but...

Q3: In your system you are assuming that the TPM is a secure chip, but the admin is malicious; as we all know the TPM does not provide security against physical access; so in your model the TPM is actually not secure?
(Adrian Perrig, CMU, United States)
A: That's one point I have not clearly stated in the threat model: we assume that cloud vendor itself is honest and would like to provide cloud computing business; vendor itself is not malicious; it's only the employees of the company who can be malicious; we assume that there are lots of physical defenses on the datacenter, like cameras, alarms on the server box -- so we assume employees do not have the right to open the server box and disturb

the TPM or install malicious hardware; he can only launch the attack from software; that's our assumption.

Q4: Can you in two sentences say where you store all of the hash blocks and how you keep them secure?
(Bryan Ford, Yale, United States)
A: The hashes are kept on disk, stored by the hypervisor. The root of the hash is kept as a Merkle tree in the memory of CloudVisor and is protected.

Atlantis: Robust, Extensible Execution Environments for Web Applications

Q1a: How does Atlantis cope with mash-ups where you're getting different bits from different domains and building them into a single site? Each one of these components has to define their own environment and what if the environments somehow conflict with one another?
(Hussam Abu-Libdeh, Cornell, United States)
A: Let's start with a simple case first. Let's say you have a single frame that's gonna be bringing. Let's take javascript code from, you know, multiple external domains. That's gonna compose in the same way that it composes now. In terms of security the Atlantis kernel is responsible for enforcing same origin policies. So the mashups should be fine. And in general mashups that should work in the standard world should work in the Atlantis world. I'm not sure if you had a particular example with mashups in mind.

Q1b: I mean…my site defines its own environment and its own markup and I'm trying to take a widget from that site and embed it into another site that defines its own environment and markup…
A: If those two sites use different frames, that's not a problem because there's strong isolation in there, they communicate through post messages or, you know, any communication interface. In terms of a single frame that can only be one definition through the high level runtime and that's whoever's the owner of that runtime

Q2a: Thank you for the very interesting and entertaining presentation. I am really with you on the high level approach...
(Bryan Ford, Yale, United States)
A: Uh oh, sounds like a trap… *[audience laughter]*

Q2b: No, no. It seems like your current implementation has some performance challenges; did you consider using instead of a high level interpreted version of java-script something like Xax or Google native client as a foundation?
A: There are really two points we wanted to make with this paper. The first is that the current "web protocol" is too big and we should use something simpler. The second is how do you architect that simple thing. Why didn't we go for native client or Xax? We are actually interested in what are those abstracts that you present to the web developer; just moving to Xax or native client wouldn't answer that question – we don't want to present "x86" as the abstraction. If browsers were to go towards exokernel-type interfaces but pick a different implementation style than what we did – that's fine – but current browsers are awful.

Q3a: Great motivation. Brian actually asked already one of my questions which is a native client and for instance which [???] to

common theorem: each webpage cover its own java script interpreter and get rid of the problems there… So I wanted to sort of ask if you have considered that this already exists. There is a technology called "Google Frame" which actually would allow you to… or "Chrome Frame" which would allow you to actually bind to the Chrome web browser and run that in IE6 and all versions of IE. So actually what the developer only has to write for Chrome, since Chrome automatically updates to its latest version. So if that suggestion is not a question well…
(Ulfar Erlingsson, Google, United States)
A: This sounds like a trap…

Q3b: The question might be: why an exokernel? Why does the fundamentals have to be small if you're only going to exercise (which Chrome Frame does) a very small fraction of the platform; even in IE6 only the networking and a couple of other primitives are used from the underline browser, the browser is very large but it doesn't actually matters since all of your abstractions are coming from Chrome Frame.
A: That's true, but I think that it's actually important in terms of system building to not want to trap this sort of generality. You can do this using this tool to bind the provided interface to the local interface. That's not quite the interface we are looking for. I think it's worthwhile to think about the fundamental types of abstractions that you actually want to present to the developer. Now in terms of things like the frame stuff, I think that work was very interesting. But I thing that one gets [???] this more fundamental issues of what should that loose level interface look like. Once again, I think that in terms of implementation, you can implement something that is somehow more natural. But I'm not sure that's the right level of abstraction. That's the same reason we didn't use the JVM, for example. We think JVM has the wrong abstraction. It's too complex and so...

Q4a: What is the actual overhead? Your graphs show the page loading times of your system, but not the time a normal browser would take to load those pages.
(Mike Piatek, Google, United States)
A: In terms of microbenchmarks…

Q4b: That looks like a very complicated graph. What I want a simple number, you loaded slashdot: how long did it take?
A: Okay, in that case, we're worse. For something like Wordpress, the overhead is about 50%. For slashdot, it's of course, much worse. It's because of the HTML parsing.

Q4c: Is it not sometimes from javaScript, rather than from the parsing?
A: Well, returning to the microbenchmarks, we can see that some of the JS implementation is a problem. For example, accessing local variables is extremely slow, but that's what we're working on fixing right now. Once we

optimize that, as well as the parsing, hopefully we will reach reasonable overhead.

OS ARCHITECTURE ----------

PTask: Operating System Abstractions To Manage GPUs as Computer Devices

Q1: What is the most optimal granularity for invalidating some portion of data blocks? I think it's a very application-specific problem. Some packets are updated and some are not [in PacketShader from KAIST] and we only want to copy some of them. I want to know what are your thoughts about that?
(Youngki Lee, KAIST, South Korea)
A: I have many thoughts. First, as I recall packet shader batches quite a bit and that introduces a larger granularity. Second, the lion's share of the overhead is transferring data across the PCI bus, PCI is not scale linearly to the size, so it would probably be cheaper to do the whole thing at one time.

Q2: How many of these abstractions are specific to GPUs and not hardware in general?
(Alec Wolman, Microsoft Research, United States)
A: If I could do a replace-all of GPU with hardware accelerator in this talk I would. The basic idea that a data block provides an abstraction for reasoning across disjoint memory spaces is fundamental and will transfer across to other accelerators. There are some "warts" in the implementation that are specific to how GPUs work.

Logical Attestation: an Authorization Architecture for Trustworthy Computing

Q1: Very cool and interesting architecture! Your webpage example left me at a loss for how practical this is. Maybe you can package up a profile picture and let the application use it, but what large-scale interesting things can it actually do? E.g., can it know how many friends you have?
(Bryn Ford, Yale, United States)
A: The principled answer is there are certain things we can't do. E.g., we can't do polls, because we can't count how people have voted one way or another on things. Getting back to the specific question, should you get to know how many pictures someone has? That's an implementation decision you make when designing your cobufs.

Q2a: Excellent document and I think that an analogy support presented is very convincing. When you presented the slide of the reference monitor, I get that the reference monitor is inside Nexus, and interposes between app and nexus: if the reference monitor fails, then the app fails. How do you guarantee the reference monitor is securely and correctly implemented?
(Paulo Veríssimo, University of Lisbon, Portugal)
A: There is no magic that it will give you secure software. TPMs give attribution, not security. So your system is as strong as your component you built on it.

Q2b: You believe that it is possible to build those standard reference monitors with 0% vulnerability?
A: Sure, I think so.

Q2c: OK, that seems optimistic.

Q3: Does other example applications you evaluated use TPM to communicate across domain boundaries?
(Ted Wobber, Microsoft Research, United States)
A: Yes, the movie player.

DETECTION AND TRACING ----------

Practical Software Model Checking via Dynamic Interface Reduction

Q1a: Your colleagues at MSR Redmond described a solution where they look at functions as modules and characterize and analyze them and use that to cut the search space. Do you think these two approaches are similar?
(George Candea, EPFL, Switzerland)
A: This idea is similar but it's just a different area. The major difference is that that work cannot discover the concurrency bugs. Symbolic execution cannot cover the concurrency state.

Q1b: That is orthogonal; they could but didn't need to. In essence you are cutting down the state space, which leads to the isolation of the explosion. Is it a similar idea in a different domain?
A: Yeah it is.

Q2: Can you explain how this works on top of Mace, which is single threaded?
(Mayasam Yabandeh, Yahoo Research, Spain)
A: For each node, it will explore a different order of concurrent events.

Detecting failures in distributed systems with FALCON spy network

Q1a: So do you think that some components that you are monitoring are more reliable than others, and some of the nodes you are monitoring might be more reliable than others, or you are operating the world by every node as homogenous and every component as homogenous?
(Emin Gun Sirer, Cornell, United States)
A: So, I understand the question had two parts. One is that these in highlight component-composed layers is one way more reliable than the other, and the other question is do you treat any layers more reliable than the other.

Q1b: Fine.
A: OK. So, we don't treat any layer as more reliable than any other. Falcon treats whichever layers I can deploy in terms of how it's monitoring. So the decision as to whether or not something is more reliable depends on the questions that, like, this wise asking. You could imagine asking more linear questions of a layer that you are expecting to not fail to offer any more strict questions, and have entire guarantees although you do expect it to fail.

Q1c: Do you think that's a realistic assumption that every component that you are monitoring is just as likely to fail as any other component?
A: I don't think that's necessarily an assumption that we are making. I said you could make that assumption by questions, like if you could use that assumption to give me a question. I don't think that there is anything fundamental in treating this node is likely to fail as this node in Falcon's approach because it's looking at this local information.

Q2: I am surprised to see this applied to Zookeper, which uses an asynchronous consensus protocol. With an accurate failure detector there's no way to detect when the cable comes out, everything is going to stop; you do not have a mechanism to detect this kind of failure?
(Robert VanRenesse, Cornell University, United States)
A: The hypervisor spy can detect that a node won't respond to pings and so this is assumed dead.

Q3a: Robert, you almost asked my question. I mean, so let me try to generalize Robert's point, inaccuracy comes in the detector due to synchronicity?
(Paulo Veríssimo, University of Lisbon, Portugal)
A: OK...

Q3b: And so, I like this a lot, but maybe something that has to come up here, is that how to insert synchronism? [not only that the local protection mechanisms because that is something I have to say that is embedded in the operating system] because if the operating system doesn't have real-time features, you cannot do synchronous things, and the networking issue is perhaps the more important one, if the network

is not synchronous, which it isn't, it is going to be difficult to say that you have the perfect accuracy of the detector because it will always be hampered by the errors in the detection mechanism if the network is gone.
A: So, one of the things I think you're getting at is this idea that the application, there is some timeout in the application that we're measuring that timeline, and that's using a real-time operating system, we will run into this issue when there's a lot of load on the operating system. What we do there is instead of measuring the time that's local within that operating system as a function of real time. We measure the actual CPU time allocated to that application

Q3c: I buy that, I buy that, even if you wouldn't have cleared that, I know that, but I'm more worried about the network, because the network is essentially asynchronous, right? So, if you don't have, for example, an alternative control network about which you can make more synchronous assumptions, you could live upon what Ethernet… You can have that problem, because the Ethernet, I mean, that network [will act asynchronously] and essentially you cannot tell slow or partitioned host from healthy host, right? I mean, I'm just asking, even accuracy drives from that in one sense, so I didn't understand how you could claim. I mean, part of the accuracy will be local, I buy it, it's neat, I like it, but part of the accuracy is given by the network, and for that you don't have control, unless you have an architecture to deal with that asynchrony.
A: So, um, the third corner case that I talked about was of network partitioning, and what we say is that because communication is required between the synchronous for accuracy, in case of network partitioned, falcon spy have to wait.

Q3d: An alternative would be that the datacenter uses an alternative network that keeps you connected for the signals.
A: Yes.

Q4: I am wondering, can you detect conditions like livelock and a sort of liveness properties? What if you have a situation where you have multiple applications communicating in the same OS and they're making forward progress but they're consuming more resources then they're supposed to?
(Marc Chiarini, Harvard, United States)
A: Interesting question. FALCON can detect livelock in an application by using the spy asking questions of the application, like "Is the application in livelock?" We had not thought about it in a group of applications before. Thank you for the question, this is something I will pursue as further work.

Q5: It seems you need to install the spies and any layer in the first step. So do you assume the developer does it manually or he just suffers that, or you need to have some alternate ways to do it?

(Ding Yuan, UCSD/UIUC, United States)
A: So this is: do you need to design a new spy or each layer that you are integrating with? So we talk in the paper about some ways to generalize the spy's architecture. So for instance, if you build a new spy for the application all you need to do is provide a function that answers the question – is the application doing what it's supposed to do? And that's the layer which we think is probably the best way to generalize the spy to alternate layers but it's definitely future work to look, at using that type of technique to sort of take this any OS-able spy.

Q6a: Just a little question about the evaluation, when you compared against the timeout-based detector, what value did you pick for the timeout?
(Rodrigo Rodrigues, MPI-SWS, Germany)
A: So, we chose a value for the timeout based on what we have seen in related work, which is generally in order of tens of seconds, and so we picked a timeout of five seconds to be aggressive with that.

Q6b: A quick suggestion, you could have been more conservative by choosing a timeout that had future knowledge of what the highest latency that you would observe in the future, and I bet you would have still looked good even though you were using very conservative value.

Secure Network Provenance

Q1: Can your system be easily deployed. In other words, can it work with traditional routers?
(Yang Tang, Columbia, United States)
A: One app is Quagga that is implemented in C++. We did not touch the source code. What we did was observe inputs and outputs and then infer dependencies based on these inputs and outputs so it is applicable to legacy applications.

Q2a: What do you use for a query language?
(Marc Chiarini, Harvard, United States)
A: That's a good question. In this work we did not look into the query language part and we have an ongoing work to look into that. For this query we execute is looking into one particular tuple change in the execution.

Q2b: So you're not doing any querying on graph structure or path expressions?
A: No, we did not have a query language that allows you specify all of that. We adopted Proql as the language but in this work we didn't explore that. Promenance is a graph and it is recursively generated.

Q3: You will have to manually define what is the correct behavior, do you have any ideas how efficiently you can do that for complex programs?
(Paarijaat Adtya, MPI-SWS, Germany)
A: For programs written in declarative languages, it is easy, for legacy applications the user must define the correct behavior. Not necessarily you must go into details, you can choose the cost to pay to get into more detail.

Fay: Extensible Distributed Tracing from Kernels to Clusters

Q1: The overhead depends on the time of the query, which type of query were the benchmarks based on?
(Mayasam Yabandeh, Yahoo research)
A: The data are all "worst case" benchmarks done by stressing the system CPU at 100% load. In the paper there are more benchmark data with real queries.

Q2a: Your system bundles computation with data recording. It supposes you know in advance the question you want to ask. Other tracing tools log the data and offer the possibility of asking whatever you want afterwards. So I am wondering if you have any intuition how common this use case is.
(Ivan Beschastnikh, University of Washington, United States)
A: You are talking about systems like flight recorder in which you try to do anomaly detection?

Q2b: Yes, that is the advantage of log data. You can use it for any kind of analytics.
A: And your question is?

Q2c: The data that you get out is aggregated. So if you want to ask a different question about the same data, it is no longer possible, you have to compute the query and run it again.
A: That is true. If you can use the knowledge that you have about the problem, you will aggregate data in a specific way. If you want to use the raw data later, than that data is gone. So, Fay is more of a dynamic instrumentation system in which you want to ask a question right now, at this point in time.

Q3a: My question has just been asked, but I'm just going to follow up on this question, asking surely you could support this kind of case by having low production in the system, would it blow up...
(Malte Schwarzkopf, Cambridge, United Kingdom)
A: Oh yes, of course, you could do that, of course. But then you run into the same kind of problems. So you could do several things. You could write a Fay tracing probe that does aggregation of the data, stores data somewhere where you don't have much overhead, and then you have to probe it in such a way that you can read it out later, that's possible, because you can write arbitrary trace code, there's nothing hindering you from just compiling a probe and put it in where it does such a thing. Umm, I was thinking about a case where you really want to just get the data immediately. So, stuff like that. This would be possible; yes, you could also do that with Fay.

Q3b: Do you have any intuition of whether it's going to blow up if you do that in a real?
A: Well, this is one thing I kind of mentioned where we are, we also implemented techniques that the flight data recorder, which is essentially a system that is storing data for a long time, umm, then when you have an anomaly, you could read it later. So, if you implement it in such a way, that's possible.

Q4a: So it seems that to do correlation across the entire cluster, since you can't change the flow of applications, essentially you're relying on their clocks to be synchronized. Is that how you correlate the different times?
(Rodrigo Fonseca, Brown University, United States)
A: I don't actually know the detail about how the events are correlated through the cluster. One way, definitely, that you could do that would be with clock synchronization, that's true.

Q4b: But that's partially true, because clocks are not really synchronized...
A: So unfortunately I don't know how that part of the application works...

Q4c: OK, we'll talk.
A: Let's take it offline.

Q5a: Thanks for doing this work, it looks like Fay can help us understand how systems function. Have you discovered any design insights with Fay that you could not have discovered otherwise?
(Yanpei Chen, UC Berkeley, United States)
A: We did. Fay was used inside Microsoft to trace some of their production systems. There are stories in the paper where they were able to use the system in order to find performance problems. For instance, in one example, we have just started to use Fay and we found that the shell had large CPU overhead just running idle. We traced this and found that it was doing a lot of interaction with the console server inside Windows. Even when the window was minimized, typing the file in the console had considerable overhead. It turned out that there were two calls all the time to check if the console was a real console compared to just typing it to a file and it was doing all this communication using RPC between the two domains. This is where the overhead stemmed from. This is one example of what we used Fay for.

Q5b: This is a specific case in which Fay's choice of tracing system calls allowed you to identify the performance problem.
A: Yes.

Scribing Sessions DAY 3

THREADS AND RACES ----------

Dthreads: Efficient Deterministic Multithreading

Q1a: I'm puzzled by this. Suppose that you know something about the execution of the threads, and you know that you don't need locks in a particular case, but still need sharing. How do you discover the synchronization?
(Sape Mullender, Bell Labs, Belgium)
A: We do not support ad-hoc synchronization, and it includes anything that depends on “racy” behavior. There was an OSDI talk on this.

Q1b: How do you know what a lock protects?
A: We don't. Every modification, whether holding a lock or not, is done in isolation, so we do guarantee atomicity but not a "correct" merge. We preserve pthreads semantics.

Q2: Since lot of them are scientific benchmarks, and some of them have considerable false sharing, what is the overhead on other types of applications, where false sharing doesn't make pthreads look bad?
(Cristian Zamfir, EPFL, Switzerland)
A: First, it is very hard to find benchmarks that don't have false sharing. There's actually a work presented in OOPSLA yesterday by one of my colleagues. But I have a slide showing you what I am talking about. I removed the overhead due to that in the benchmark slides, and you can see we are just marginally slower than pthreads in that case.

Q3: It might be in the paper, but in general I was wondering if you did anything about concurrent synchronization primitives?
(Jean-Philippe Martin, Microsoft Research, United States)
A: When you run these operations in isolation, you don't get the performance you were expecting. We don't support such ad-hoc synchronization, but it doesn't break the program. We just lose the performance benefits.

Efficient Deterministic Multithreading through Schedule Relaxation

Q1a: It seems to me that you really need to have the input of you program before you start running it. What happens if you need to decide afterwards?
(Cristian Zamfir, EPFL, Switzerland)
A: In our system, you don't need to read all the input in the command line at one time. It can take continuous input. Our previous work has solved this by taking the input as a decision tree to determine the schedule.

Q1b: Overhead proportional to the number of memory accesses to check if you can reuse the schedule?
A: Actually, that's not the case. We have our own definitions of common functions like 'scanf', etc. Those read the commands not from continuous input stream.
A (by Junfeng Yang): Since we use this slicing, we can slice out a lot of the data and only check the data that is part of the constraints. If it is not part of the constraints, you don't need to check any of it.

Q2: If the first schedule you record happens on a loaded system, this schedule might be sub-optimal. Then it will get replayed. Optimal schedule depends on system load, for instance the schedule might be serial if the system is really loaded. How is the performance affected by the initial recorded schedule?
(Antonios-Kornilios Kourtis, ETH Zurich, Switzerland)
A: Schedule could be big. So we compress the schedule. Schedule only contains race conditions, not non-race conditions. Schedule only contains small number of events.

Pervasive Detection of Process Races in Deployed Systems

Q1: Do you introduce additional locking for recording?
(Bryan Ford, Yale, United States)
A: Yes, we introduce some extra locking, but it's not a problem because we can run the recorder all the time on the production system, so we would never hit unexplored execution paths.

Q2: What is your definition of a race? I wouldn't necessarily call it a race, since a race is only something you don't want. Are you not just detecting non-deterministic execution?
(Marcus Aguilera, Microsoft Research, United States)
A: So the question is, what do you mean by race? Races are executions that cause a program to fail after a certain period of time. We want to detect those cases where bugs manifest themselves due to non-properly synchronized races.

Q3: What is the granularity of the recorded objects?
(Yubin Xia, Parallel Processing Institute, China)
A: It depends. We only track the objects exposed to user space. We consider branches of resources. For example, if a file is read or written by a file, we don't take a lock on the whole file. We look at the piece of data that each process accesses separately.

Q4a: Who decides what is harmful race?
(Cristian Zamfir, EPFL, Switzerland)
A: Detection of a harmful race is as good as our checker. Our checker determines what a harmful race is.

Q4b: How do you check the race for /ect/passwd for instance?
A: Here you have to run a custom checker. You cannot just use the built-in one. You can for example use linearized run and check if the two executions did something completely different. This is some future work: how we can have really good checkers.

Detecting and Surviving Data Races using Complementary Schedule

Q1: This is a follow-up on the two questions from the previous talk. What's a race, and what's a harmful race. You were more specific, but I think both of your definitions were wrong. You say that state divergence…
(George Candea, EPFL, Switzerland)
A: I say a data is harmful, because... A harmful is what we care about. If multiple executions with reordered instructions, and it turns out that it doesn't violate the functionality of the program then not harmful. But a problem of data analyzers is that they cannot detect the races we do. The conclusion that it's a harmful race is incorrect. You'll see that frost have both harmful and benign threads. Gives an example using a master and worker thread. The master doesn't care which thread finished first. Frost will say it's a harmful race, even though they're benign. We do drastically well, however, over traditional tracers,

GEO-REPLICATION ----------

Transactional storage for geo-replicated systems

Q1: First of all, this is really nice stuff and geo replication and asynchronous are the way to go. So the two comments I have are: One, we actually have a proof that the PSI protocol is just about the best you can do in a geo replication setting. Second is too bad you chose 'csets' because they have anomalies. There are better sets to choose from.
(Marc Shapiro, INRIA & LIP6, France)
A: 'csets' is just one set, it's good because it doesn't have write-write conflicts.

Q2a: How reasonable is the assumption that one can come up with counting set approach and avoiding write-write conflict and I'm going to assume that its feasible to come up with this. My question is that, what is specific about the wide area network? Why cannot you apply the same idea to single data center node?
(Mayasam Yabandeh, Yahoo Research, Spain)
A: Ok, excellent question. So, I think the short answer is that, within a single data center, I believe, and other people have demonstrated it is possible to achieve snapshot isolation, and if this is possible, why go for a weaker model? So I think this is the short answer.

Q2b: But, does it sound weaker in single node data center? Does it? Because the counting set approach seems to require more effort to get it right.
A: But the application programmer does have to deal with the kind of slightly strange semantics of the counting set.

Q2c: So, its not weaker but harder.
A: If you can provide a perfect set with snapshot isolation within one single data center at a reasonably low cost, you might want to do this, so it is really a tradeoff between the wide-area and [...], with slightly stranger sets in exchange for performance.

Don’t Settle for Eventual: Scalable Casual Consistency for Wide-Area Storage with COPS

Q2a: From an abstract point of view I see lot of similarities between your model and snapshot isolation:
1) Both of you talk about maintaining multiple version of data.
2) Both of you talk about snapshots
3) Both of you try to detect write-write conflicts
I wonder about the differences?
(Mayasam Yabandeh, Yahoo research)
A: Are you asking the difference between this and what Jinyang talked about, which is snapshot isolation in general?

Q2b: I want to know in general.
A: So, in general, snapshot isolation is a database consistency property. It is a stronger consistency than what you get with our system; with snapshot isolation you can do all these transactions with read and writes, we don't have that in our system; what we have in our system is that we guarantee low latency, you will always complete right away.

Q2c: From an abstraction point of view I want to compare your method to a snapshot isolation method.
A: This is sort of tricky. In the last talk, Jinyang had this spectrum of consistency models; what she was showing was more from the database side where you have these transactions and involve lots of different keys at a time, lots of different updates and operations. We are more from the shared memory side where all these things involve one operation at a time; how exactly these consistency models interact is a more complex graph; I would say the snapshot isolation is a stronger consistency model, but we have better performance.

Q2d: So, you give it a different name, but they seem like transactions.
A: So, we only have read transactions and you can only read multiple values in a transaction. So we can have read-write conflicts in our systems [...] or we have to use an application specific function that is going to resolve these conflicts.