Session 8: Cloud Networks 2.0

Transcribed by: M. Taimoor Tariq, Ammar Tahir (UIUC)

Computation-Centric Networking

***Yuhan Deng**, Angela Montemayor (Stanford University); Amit Levy (Princeton University); Keith Winstein (Stanford University)*

Paper makes an argument for the new model of service for cloud. Cloud should guarantee completion of computation instead of availability of resources for some unit time.

Questions:

1. **Scalability of data?**
   If we are able to tell the size of the dataset, we can decide whether to move data to computation or other way around.
2. **Why are you trying to do fine grained tracking**
   Knowing which files are actually touched can have substantial savings
3. **Is it useful for language runtime to provide such visibility? E.g. API to see what kind of data is moving through pipeline**
   We picked assembly because most high level languages can be translated to assembly, but it is a useful suggestion.

Rethinking Cloud-hosted Financial Exchanges for Response Time Fairness

***Prateesh Goyal**, Ilias Marinos (Microsoft Research); Eashan Gupta (Microsoft Research, UIUC); Chaitnya Bandi (Microsoft Research); Alan Ross (Microsoft); Ranveer Chandra (Microsoft Research)*

For financial exchange systems working in the cloud ensuring fairness in terms of delivery of information is critical. This paper argues rethinking design and centering discussion around response time fairness.

Questions:

1. **What about in the case of non-closed systems?**
   External data streams are not as important
2. **What is the guarantee that Microsoft is not colluding**
   Auditing can help
3. **Why don't you simply take the hash of the information and here's the thing I am responding to? Is there incentive for me to lie?**
   The users might not be honest so we cannot rely on them. In some cases it might make some sense for a dishonest user to pretend to falsify a timestamp.
4. **What tells me as one of the players that MSR isn't colluding with another one of the players?**

There could be some sort of Auditing system. The participant can actually see the activity sequence transparently so the user can see if the provider was colluding with someone.

Load Balancers Need In-Band Feedback Control
***Bhavana Vannarth Shobhana**, Srinivas Narayana, Badri Nath (Rutgers University)*

Load balancers must be server performance aware. How? Measure user response latency.

Questions:

1. **What if you put LB in the P4 switch?**
   If direct server return is not enabled, it can infer much more stats about the server.
2. **In all your examples you are assuming you do have an LB. What if there is no LB or it fails and you have to migrate?**
   Every LB will be receiving the traffic and even if they are spread across DCs. And even if one goes down, there will be other LBs directing traffic to the same servers. So essentially replication keeps it safe.
3. **Assumption is that clients are close to a DC. in a larger scale deployment, the responses that you send get delayed a lot and the LB gets idle information. So it would be better for the servers to just respond and share whatever load they have.**
   Our assumption was that all clients were close to a DC. Currently we are evaluating our traces online traces with more variations in the traffic patterns

---

End of Session Discussion

1. **Traditionally HFTs have been reluctant to be involved in anything unrelated to physics. Do you have a proof that game doesn't change if you move to this design**
   It depends on the algo you're using, we provide thot=retical guarantees on response time fairness. There is work that looks at fairness which is not even related to speed but our focus on this work is on speed.
2. **How can generic compute centric programming be? Can we express loops? Whether data is mutable?**
   We are even more flexible than Spark, you can do loops. If all computes are reproducible it is okay to mutate data.
3. **Motivation of moving financial exchanges to the cloud?**
   One reason is cost, and scalability. 1000s of participants should be supported. Since these participants are just running algorithms, it is manageable to do it in the cloud.
4. **Parallel pipelining between I/O and compute, would that be helpful.**
   It is orthogonal to our work.

You can integrate multiple sources of data but it will increase the latency a lot.

5. **Is this work (financial exchange paper) creating an incentive to do everything immediately or make systems very fast?**
   We show in paper that it is always in your best interest to respond even faster.
6. **What if my hardware is worse (for financial exchange paper), am I at disadvantage?**
   Yes
7. **There is a general trend of financial companies to the cloud. So is it possible that now even ordinary people like ordinary people can join the high speed trading area as well?**
   Yes, currently, the set for which people can be allocated high speed and speed time fairness is limited, that is one of the motivators. Its also more about having a better algorithm rather than just better speed.
8. **Goal is to detect if an intrusion affects the network performance. Outputting the track record. If there could be an explosion of data.**
   Not really trying to output data, trying to measure system level calls or something.
9. **All these problems seem to be related to best effort delivery models. And seems to be an unfortunate consequence of how the internet works and was ported to DC. Have you thought about going away from the best effort deployment model and wanting stronger semantics from the best effort delivery model. What could help us solve these probs**
   For fin ex, if you can guarantee that time bounds are guaranteed, you can use techniques like bandwidth reservation, such properties might help out
   Also, the network is only part of the problem in terms of variation we are trying to address. There are a lot of other networks outside as well such as software, layers of virtual stack, factors beyond best effort as well. Fundamentally, you still have the issue that machines have application software that they are running for shorter and shorter time so guaranteeing their performance is harder and harder.
   You can make your network do more predictable things but that will throttle the performance and so at the end of the day the best effort might just provide you with better performance.
   Some sort of result that best effort is better at some level, doesn't necessarily have to be better in all cases.
10. **If the traffic is sent back to the LB, then it's easy but if we are going around the LB. Why don't we create a . Load balancer should take over the session from client and terminate tcp conn**
    The problem is easier if you visibility into both directions of the traffic. But that is not a config you can run for all traffic e.g. DDOS. It is preferable to have an L4 load balancer followed by a L7 balancer. The scheme you suggest cannot work on scale and is very expensive.