Cloud Networking: current trends, problems and some solutions

Ahmed Mohamed Abdelmoniem Sayed

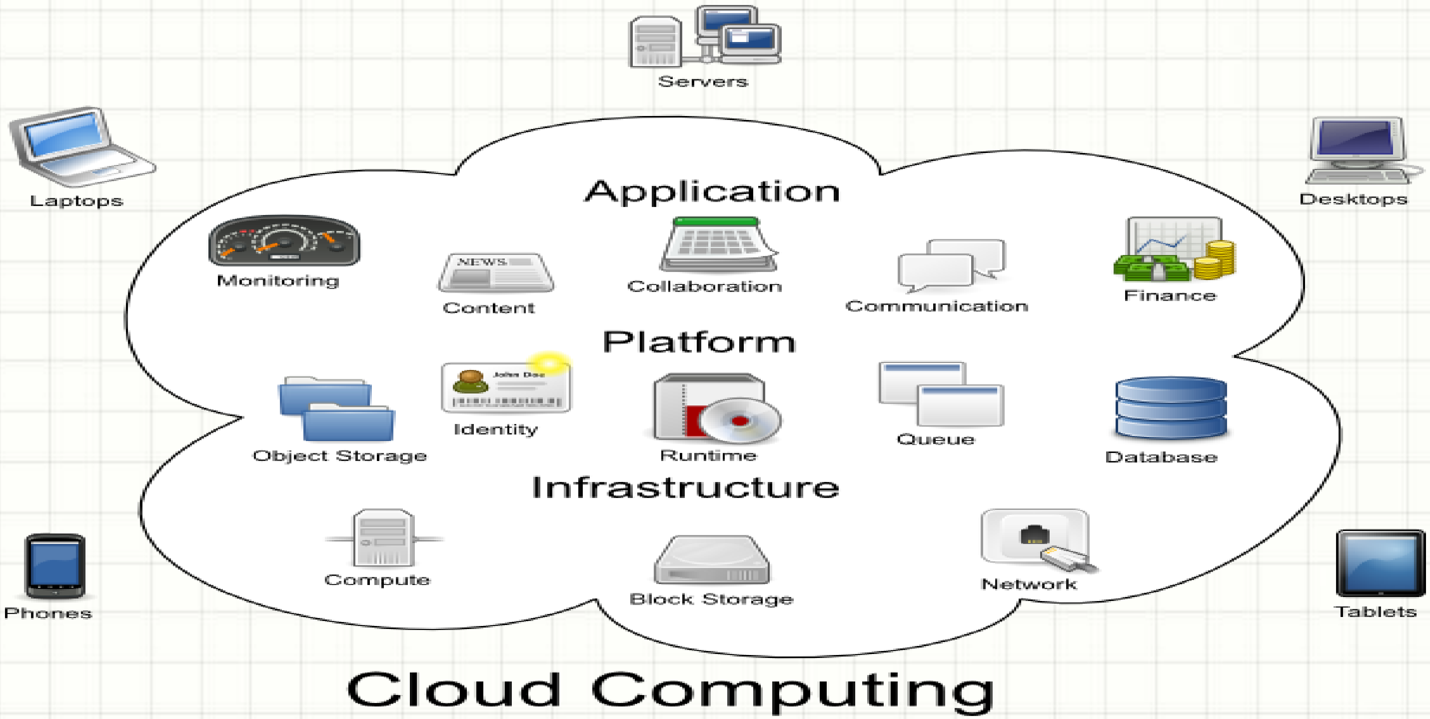
PhD Student at CSE Department
Hong Kong University of Science and Technology

Outline

1. Introduction
2. Data Center Networks
3. Internet Congestion Control
4. DCN Congestion Control
5. Our work & Future Works

Cloud Computing Era

- **Cloud computing** is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like the electricity grid.

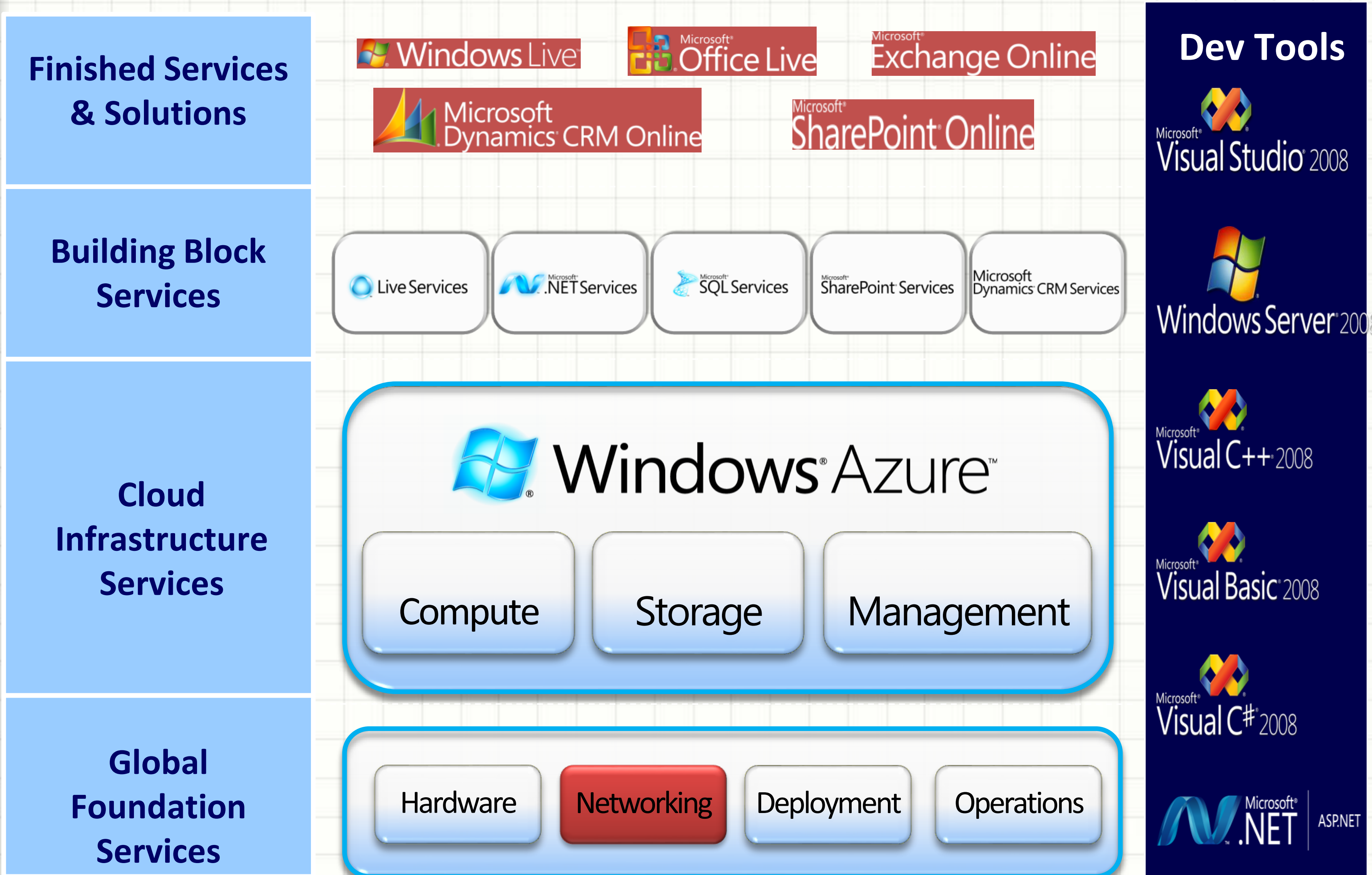


*source : *Wikimedia Commons – Cloud Computing*

Why Cloud Computing?

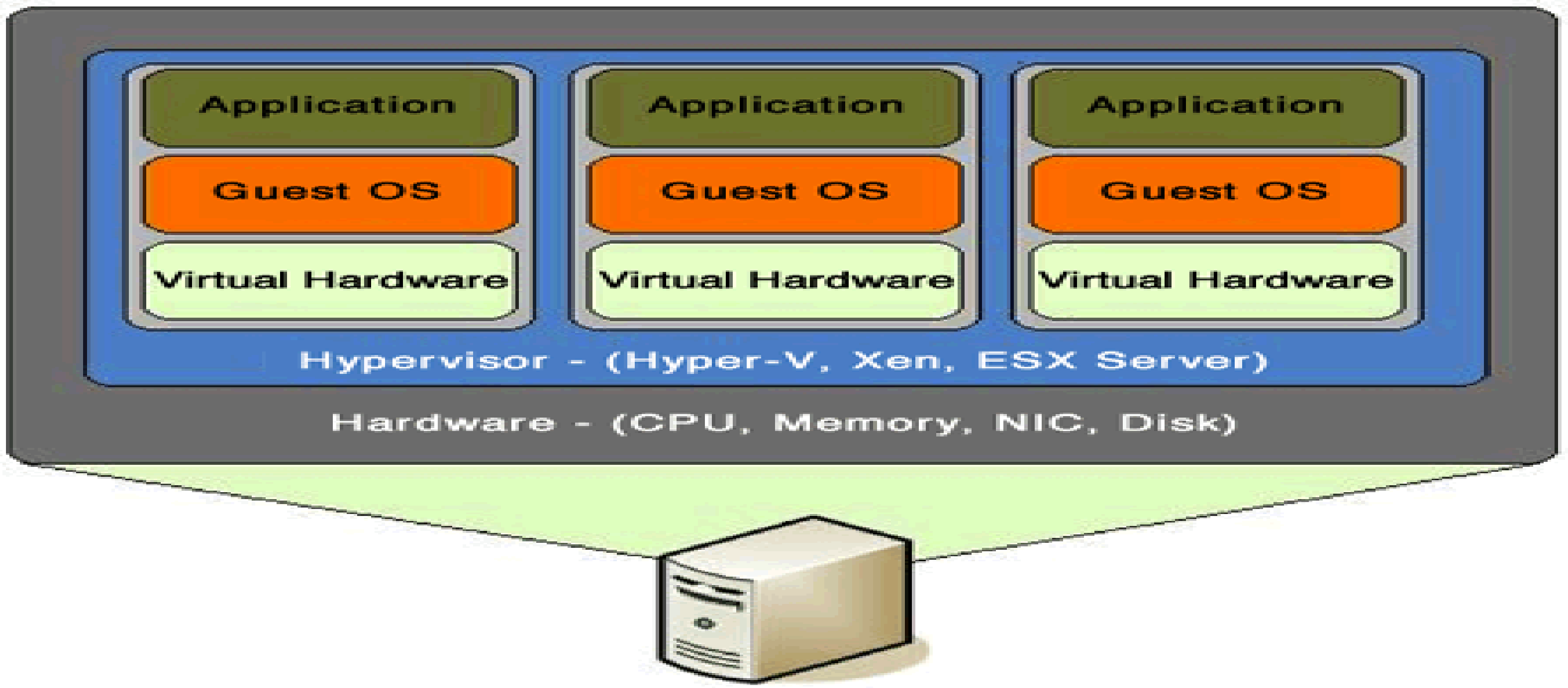
- Elastic resources
  - Expand and contract resources
  - Pay-per-use
  - Infrastructure on demand
- Multi-tenancy
  - Multiple independent users
  - Security and resource isolation
  - Divide the cost of the (shared) infrastructure
- Simplify app deployment & management
  - Common programming model across mobile, browser, client, server, cloud

Microsoft's Cloud Platform



*Source: http://research.microsoft.com/pubs/102318/Location-based%20service%20on%20the%20Cloud.pptx

Virtualization

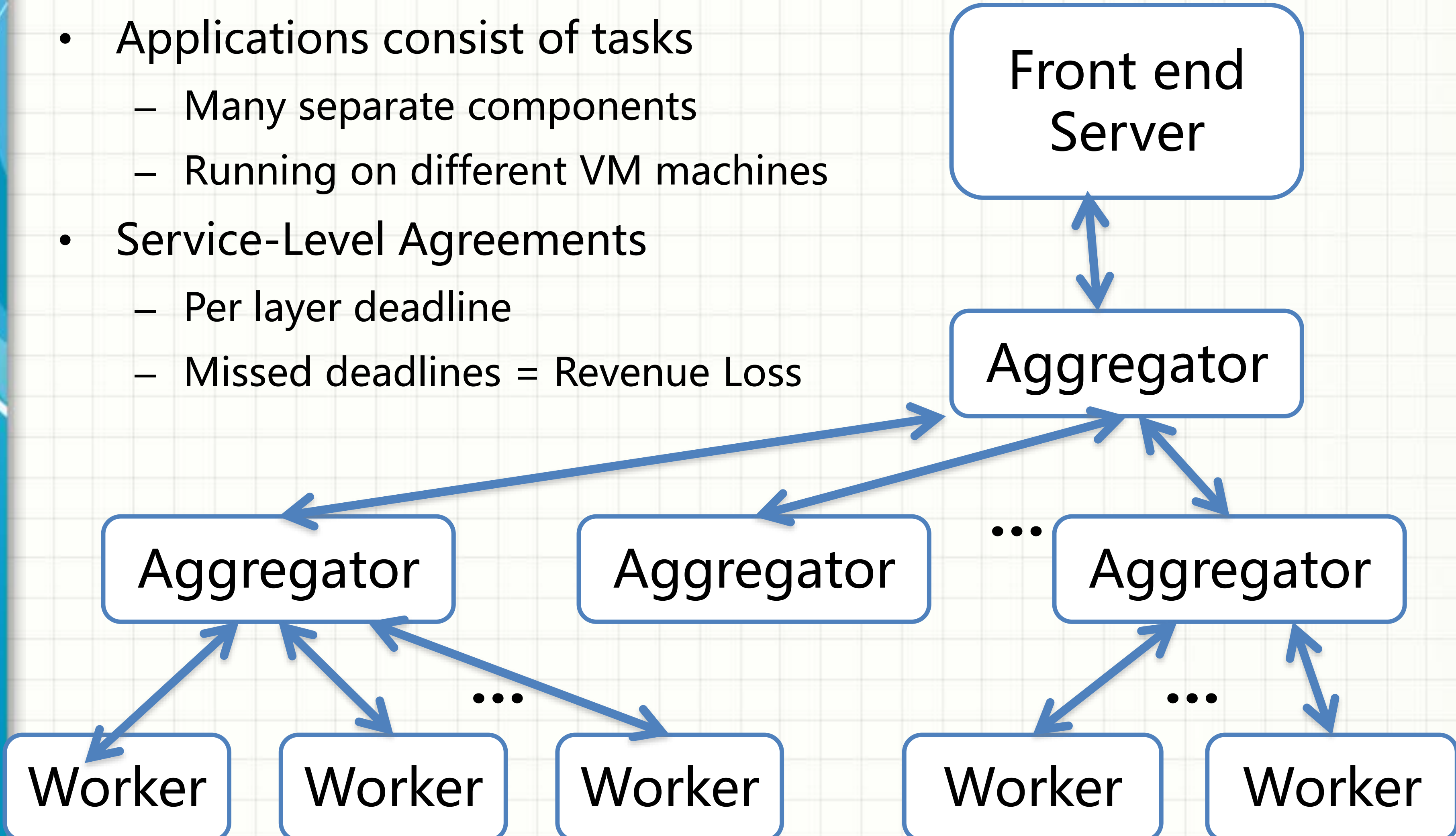


*source : *Wikimedia Commons – Public Domain*

- Multiple virtual machines on one physical machine
- Applications run unmodified as on real machine
- VM can migrate from one computer to another
- Each VM is typically owned by a tenant in public DC

Multi-Tier Applications

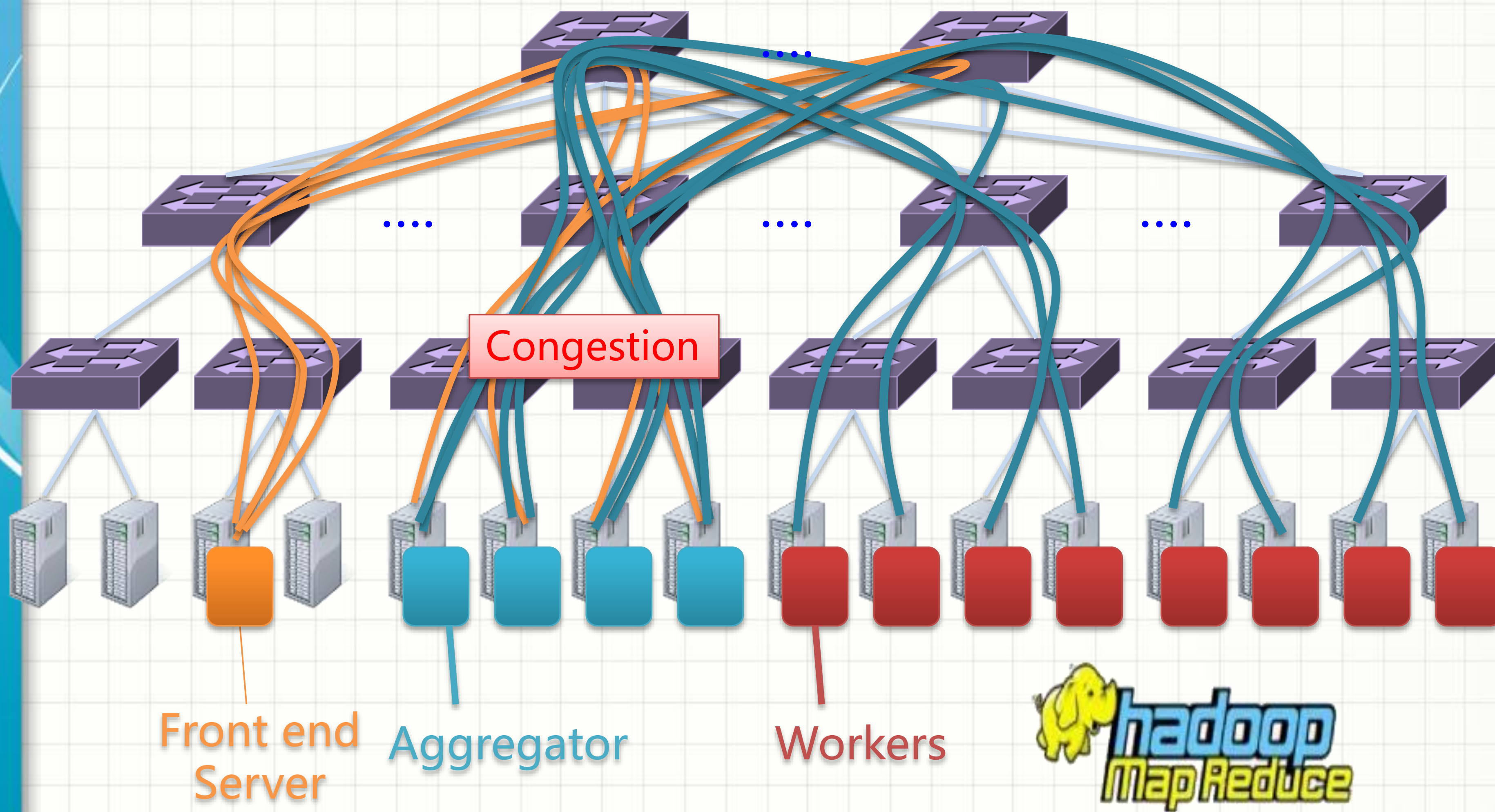
- Applications consist of tasks
  - Many separate components
  - Running on different VM machines
- Service-Level Agreements
  - Per layer deadline
  - Missed deadlines = Revenue Loss



Outline

1. Introduction
2. Data Center Networks
3. Internet Congestion Control
4. DCN Congestion Control
5. Our work & Future Works

Applications inside Data Centers



Gmail, Bing, Dropbox, ...

The Need for Reconciliation

- Partition/Aggregate is the foundation for many large-scale web services (e.g Google Search, Facebook Queries)

- Query [1KB-100KB]
- Short messages [100KB-1MB]

(Coordination, Control state)

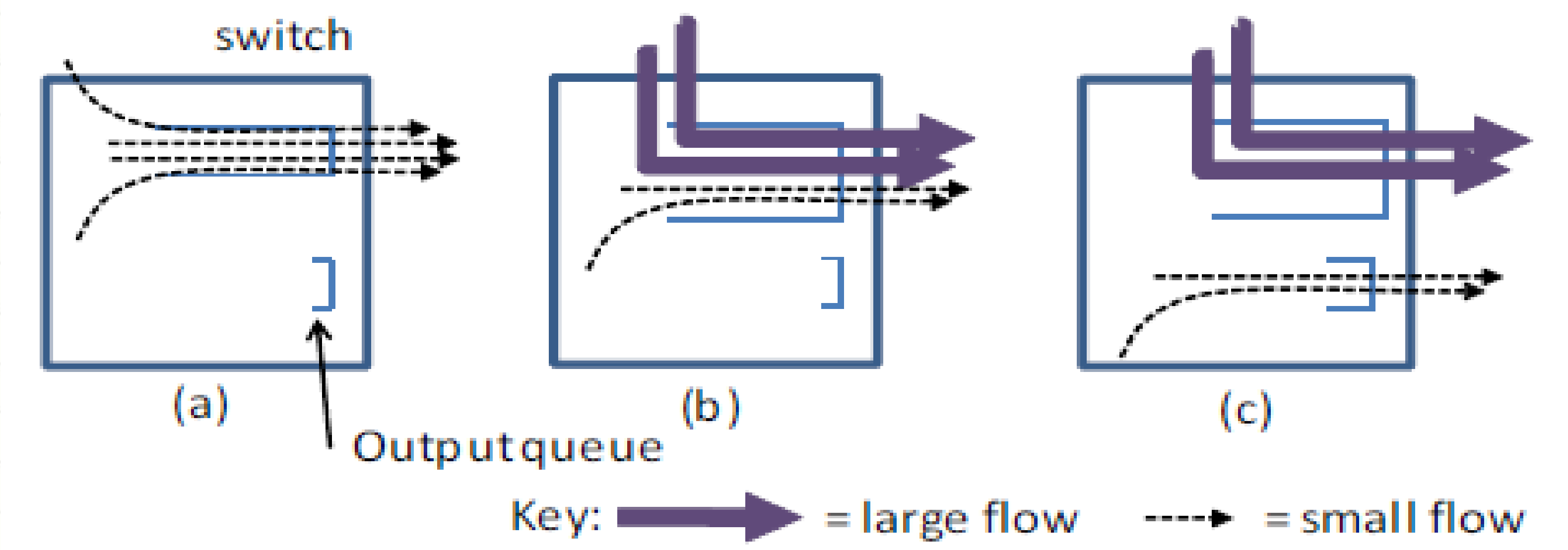
- **Delay-sensitive**
- **Large in number**
- **Few bytes amount**

- Large flows [>1MB]

(Data update, VM migration)

- **Throughput-sensitive**
- **Few in number**
- **Large bytes amount**

Typical Sources of Performance Degradation in Data Center Networks [4]



a) Incast : many flows go through the same port within a short interval ➔ The buffer space get exhausted ➔ packets of some flows dropped ➔ miss deadline
b) Queue buildup : even with no packets are dropped ➔ short flows experience increased latency queued behind packets from the large flow
c) Buffer Pressure: when shallow buffered switch (shared memory) is used ➔ short flows on one port to be impacted by activity of long flows on other ports.

Outline

1. Introduction
2. Data Center Networks
3. Internet Congestion Control
4. DCN Congestion Control
5. Our work & Future Works

TCP Congestion Control

- Designed to address Internet congestion problem
  - Window-based (AIMD) adjustment of sending rates.
  - Assume packet losses → network congestion
  - many variants: Tahoe, Reno, Vegas, Cubic, Westwood, ..
- Router assistance to TCP
  - Random Early Detection (RED) : measures congestion based on weighted moving average of queue length and either drop/mark probabilistically
  - Explicit Congestion Notification (ECN) : is used for conveying congestion information to the senders
- Clean-slate approach
  - eXplicit Congestion Control (XCP): Congestion Window + Feedback (in ACKs)

Differences Between DCN and Internet/WAN

Characteristic	Internet/WAN	DCN
Latencies	Milliseconds to Seconds	Microseconds
Bandwidths	Kilobits to Gigabits/s	Gigabits to tens of Gbits/s
Causes of loss	Congestion, link errors, …	Congestion
Administration	Distributed	Central, single domain
Statistical Multiplexing	Significant	Minimal, 1-2 flows dominate links
Incast	Rare	Frequent, due to synchronized responses

Outline

1. Introduction
2. Data Center Networks
3. Internet Congestion Control
4. DCN Congestion Control
5. Our work & Future Works

Data Center Transport Requirements

1. **High Burst Tolerance**
   – Incast due to Partition/Aggregate is common.

2. **Low Latency**
   – Short flows, queries

3. **High Throughput**
   – Continuous data updates, large file transfers

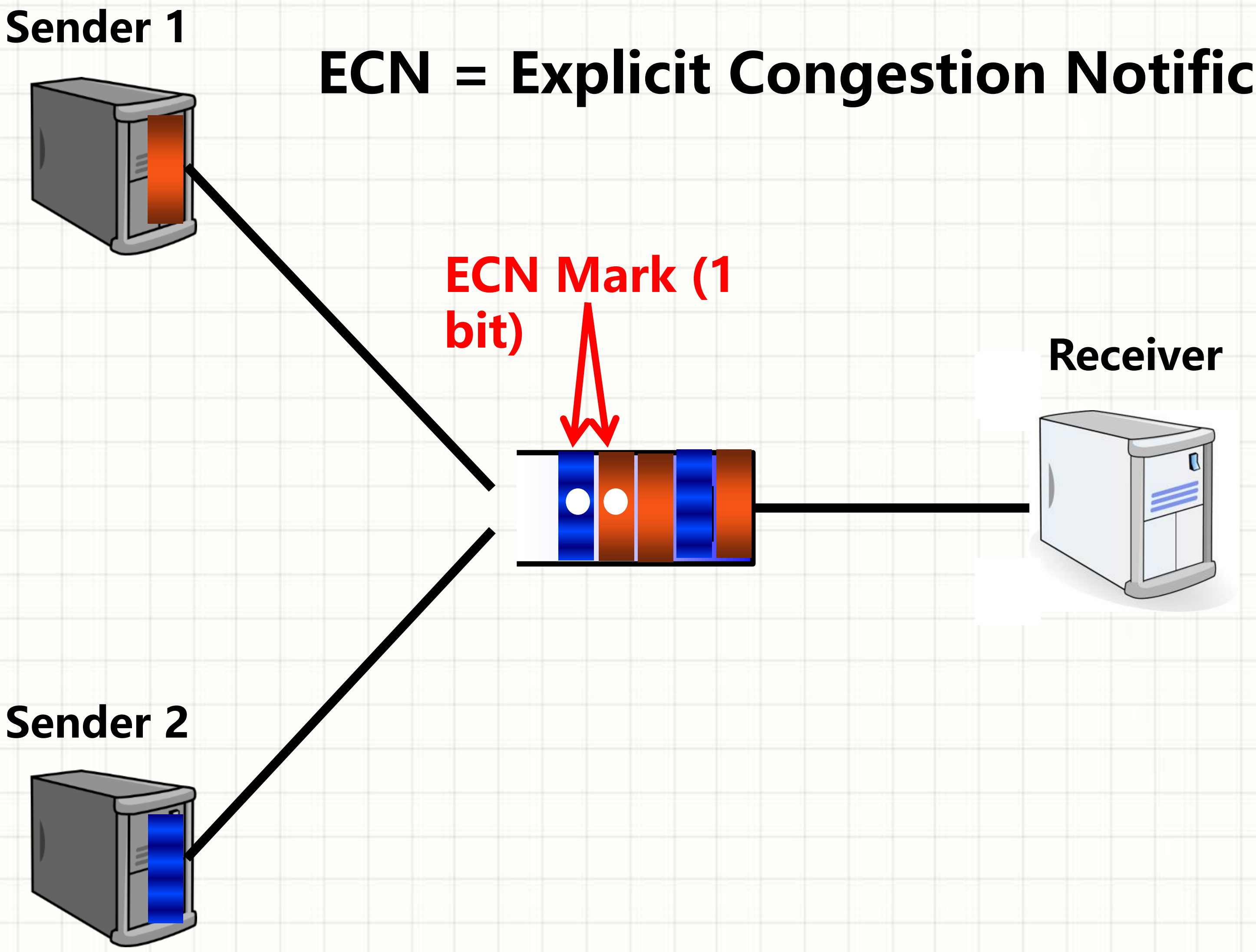
The challenge is to achieve these three Conflicting Requirements

Existing Solutions

1. Sender-Based :
   – Mirco-seconds MinRTO [3] and **DCTCP [4]**
2. Receiver-Based :
   – ICTCP [10] and PAC [11]
3. Switch-Assisted :
   – PFabric [7] and Cutting-Payload [12]
4. Deadline-Aware :
   – D^3 [5] , D^2TCP [8] and PDQ [9]

Data Center TCP (DCTCP)

ECN = Explicit Congestion Notification



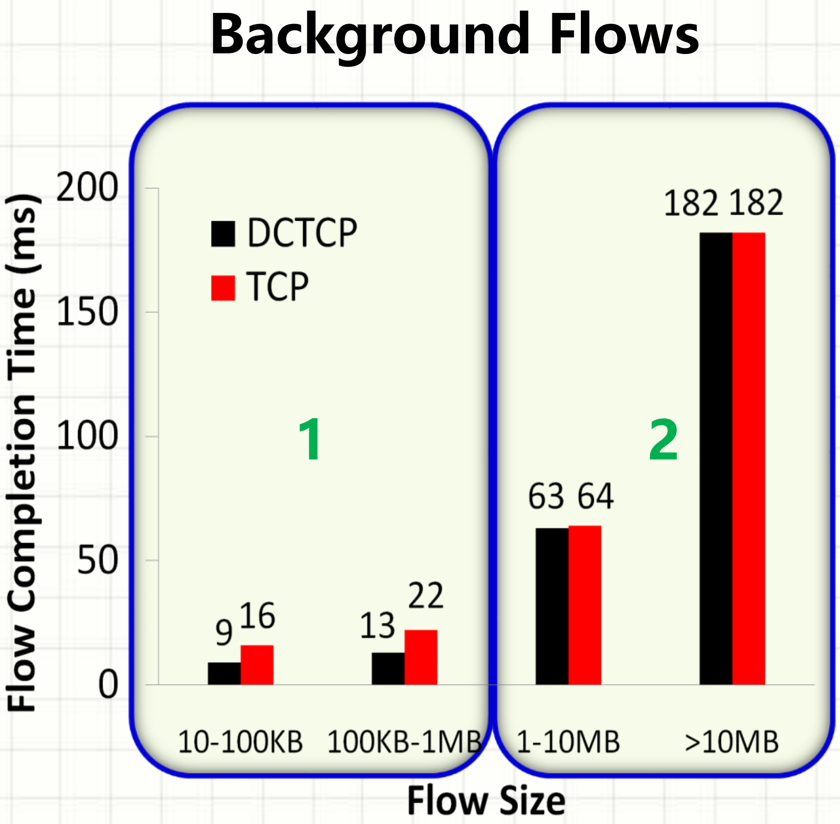
Two Key Ideas

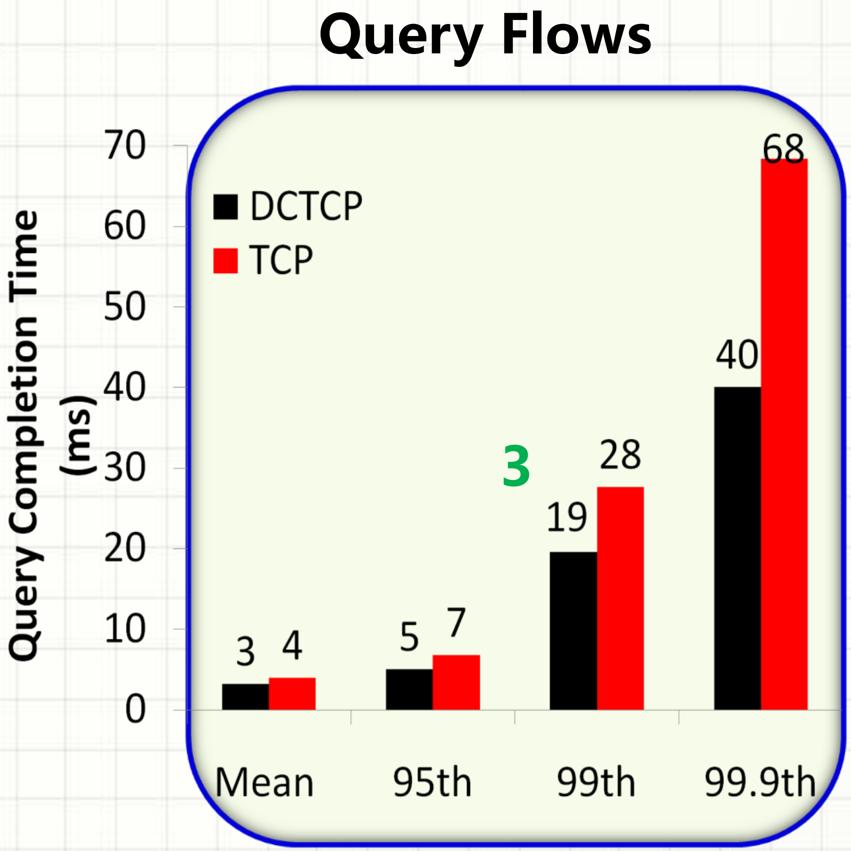
1. React in proportion to the **extent** of congestion, not to its **presence**.
   - ✓ Reduces **variance** in sending rates, lowering queuing requirements.

ECN Marks	TCP	DCTCP
1 0 1 1 1 1 0 1 1 1	Cut window by 50%	Cut window by 40%
0 0 0 0 0 0 0 0 0 1	Cut window by 50%	Cut window by 5%

2. Mark based on **instantaneous** queue length.
   - ✓ Fast feedback to better deal with bursts.

DCTCP Cluster results





1 - Low latency for short flows.
2 - High throughput for long flows.
3 - High burst tolerance for query flows.

DCTCP Summary

- ✓ Handles bursts well
- ✓ Keeps queuing delays low
- ✓ Achieves high throughput
- ✓ Based on ECN, a mechanisms already available in Silicon.
- × Can not handle incast of very large number of senders
- × Limited by the lower bound on window size
- × Requires modification to sender and receiver TCP stack
- × Fine-tuning of switch parameters
- × Not suitable for public data centers

Outline

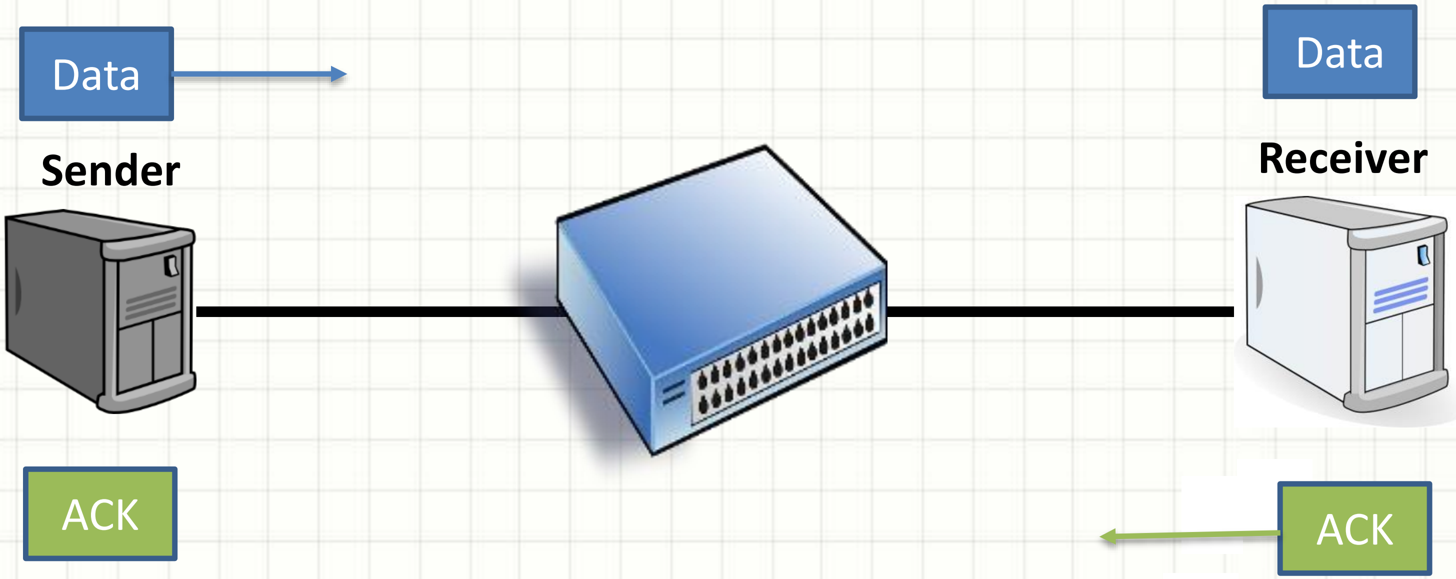
1. Introduction
2. Data Center Networks
3. Internet Congestion Control
4. DCN Congestion Control
5. Our work & Future Works

Our Work

- Simple yet efficient switch-assisted solution
- No modification to the TCP sender or receiver stack.
- Solution that fits in regardless of TCP flavor.
- Appealing to public cloud operators.
- Incremental deployment is possible.
- IQM [12] at Globecom15
- RWNDQ [13,14] at Cloudnet15 and IPCCC15

TCP Flow Control is the answer

Flow Control is part of all TCP flavors



- TCP header has a Receive Window Field which is a major part of TCP’s rate control (sending rate).
- Send Window = Min (Congestion Win, Receive Win).
- Hence, No modification is required to TCP

IQM - Two Key Ideas

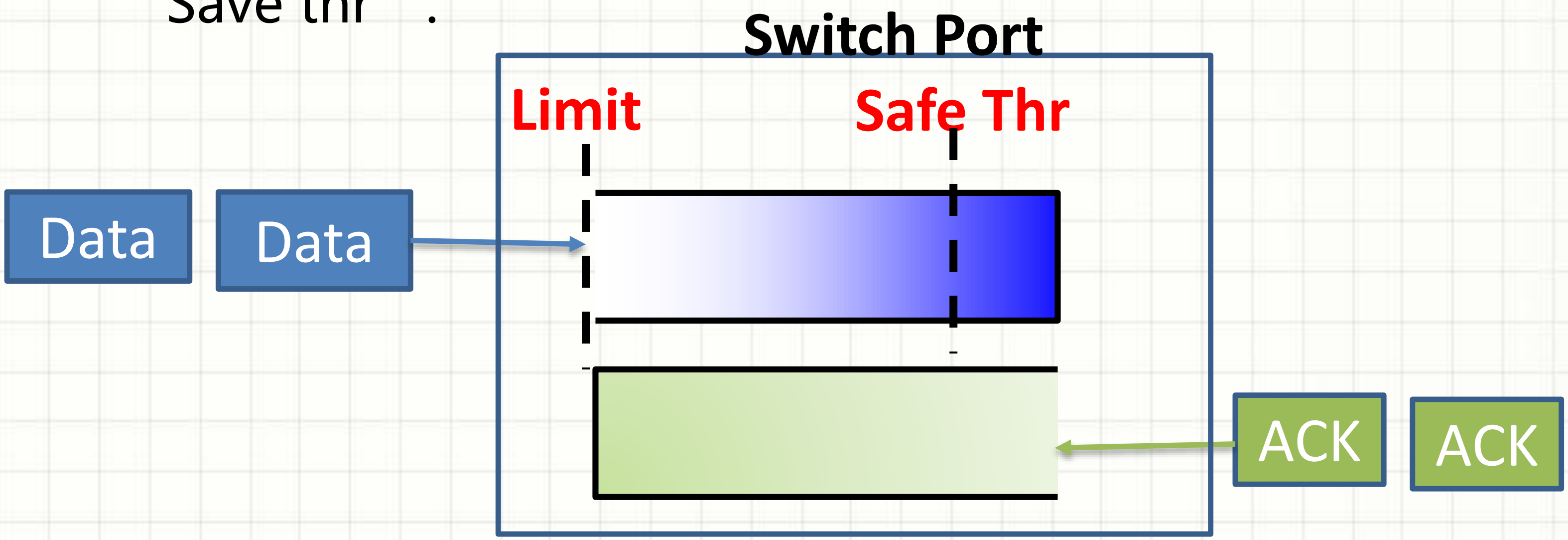
1. Switch port toward destination monitors connection setup rate.
   - ✓ Count the number of SYN-ACKs and FINs.
   - ✓ The difference represents the expected new connections.
   - ✓ If expected number will overflow buffer → incast flag.

2. Set TCP receive window to 1 MSS during Incast.
   - ✓ Proactively react to possible incast congestion event.
   - ✓ Clear the buffer space occupied by elephants.
   - ✓ Make room for the incoming incast traffic.
   - ✓ Disable rewriting when incast event clears.
   - ✓ Low computation and rewriting overhead.

IQM Algorithm

Switch side (Continuously monitor incoming SYN/FIN):

- If extra traffic > "limit" → raise incast flag.
- Set TCP RWND=1 MSS during incast epoch.
- Disable window rewriting when the queue drops back to "Save thr" .

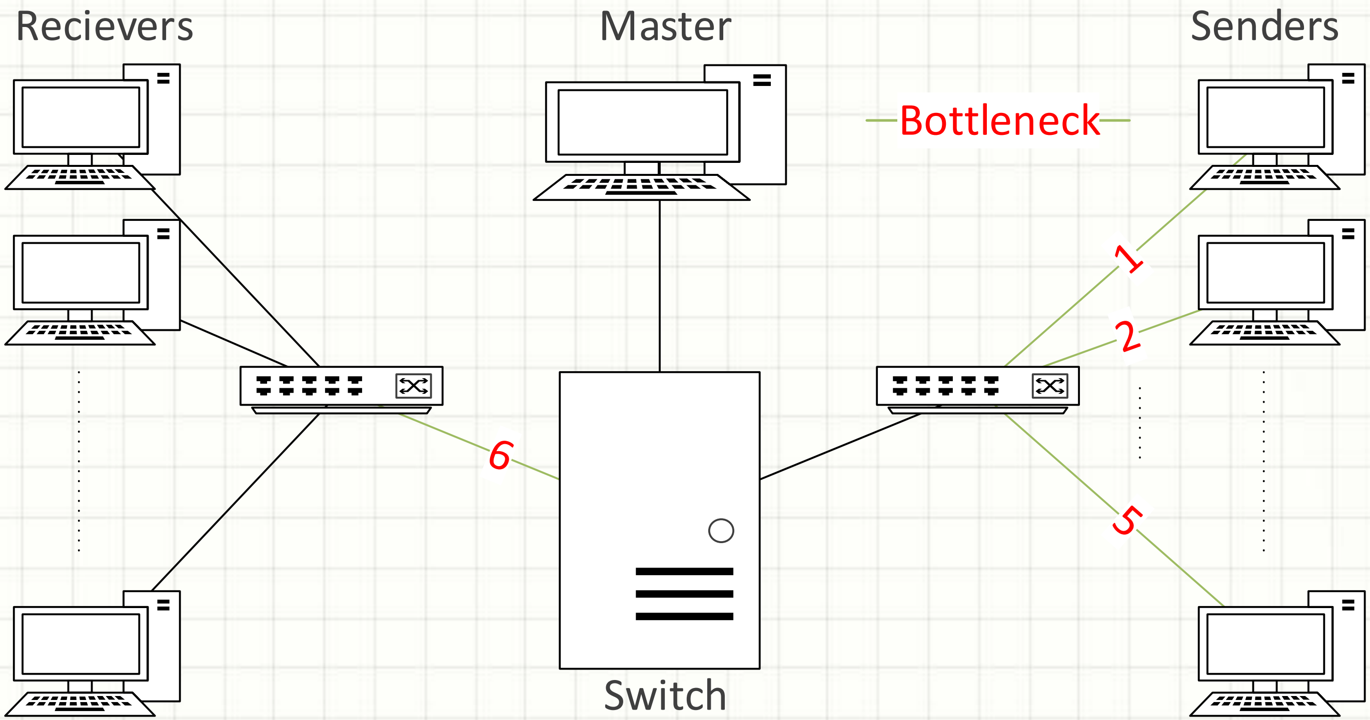


Sender and Receiver side (No Change):

Send Window = Min(Congestion Win, Receive Win)

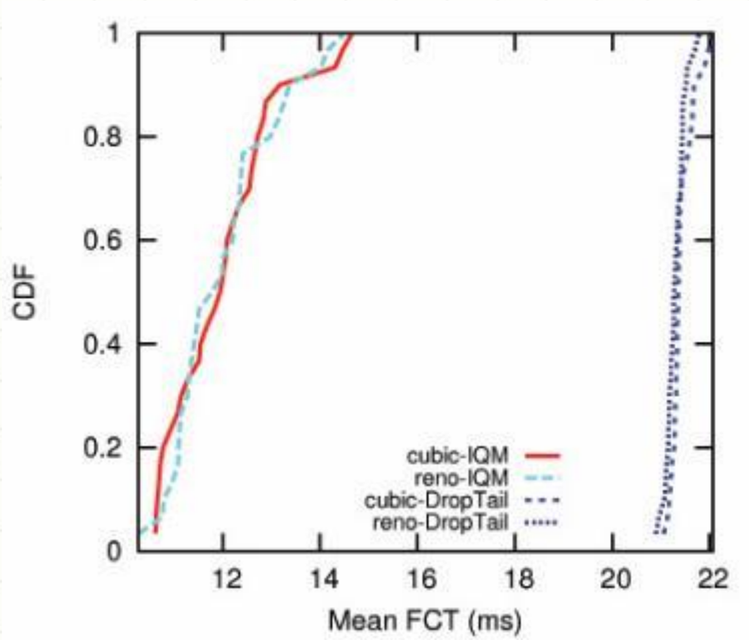
Testbed Setup

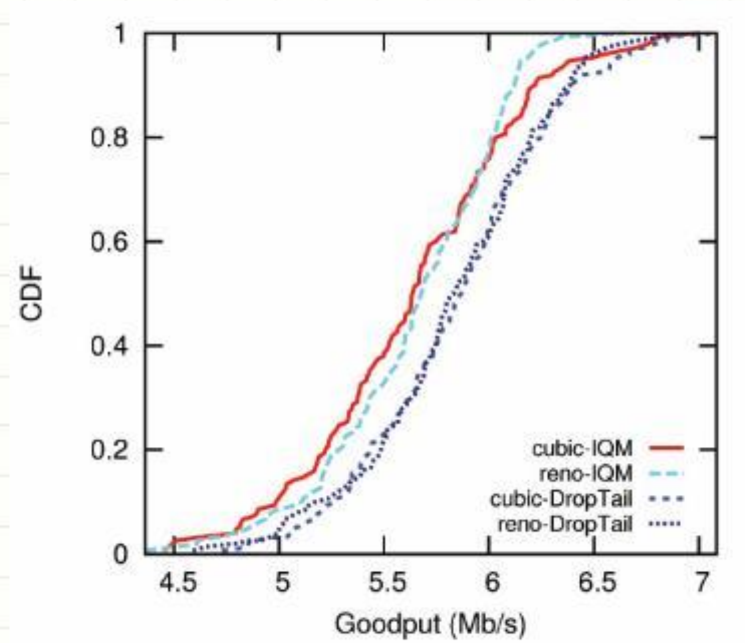
- 12 servers: 1 master, 1 OVS physical machine, 5 senders and 5 receivers with OVS for the vPorts.
- Mice flows are Web page requests of 11.5 KB.
- Elephants flows are iperf long lived connections.



Sample - Experimental Analysis

- Small Scale Testbed using Open vSwitch
- Scenario depicting 150 elephants against 30 Mice.
- Mice Goal: Low Latency and low variance.
- Elephants Goal: High and enough throughput





RWNDQ - Two Key Ideas

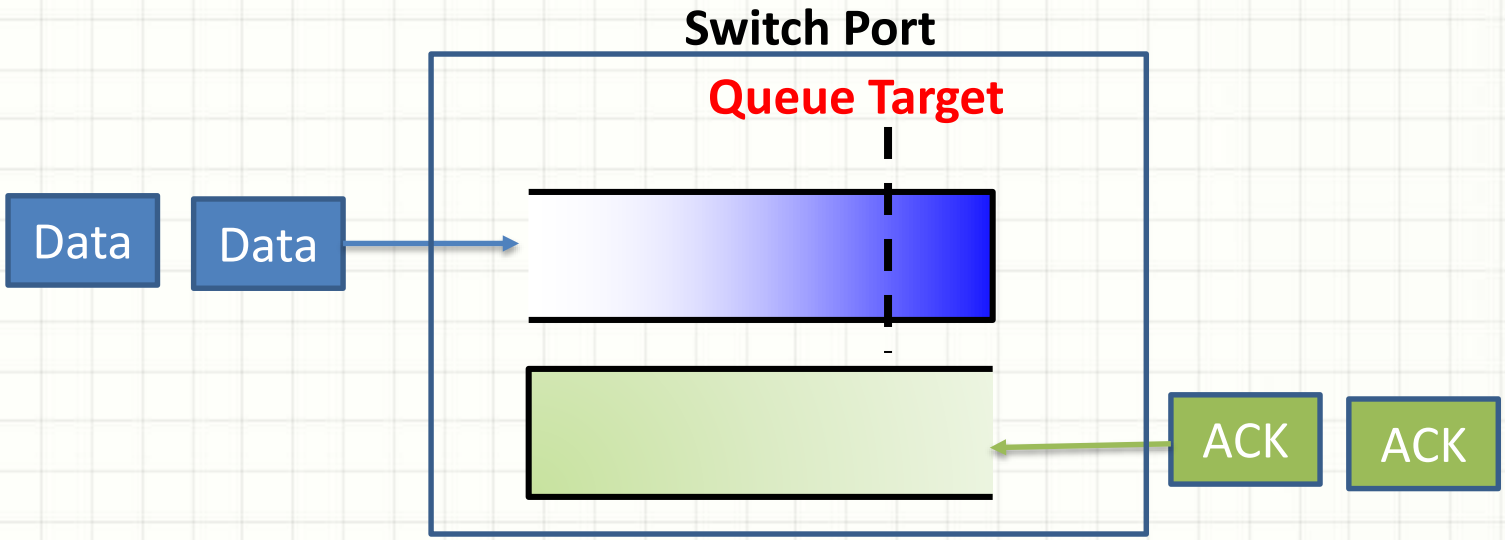
1. Switch egress port toward destination is a receiver of the data.
   - ✓ Buffer occupancy change over time
   - ✓ Buffer occupancy reflects level of congestion.
   - ✓ Locality of number of ongoing flow information.

2. Send explicit feedback by leveraging TCP receive window.
   - ✓ Similar to XCP and ATM-ABR techniques.
   - ✓ Receive window controls the sending rate.
   - ✓ Feedback is less than ½ RTT away.
   - ✓ Fast reaction to congestion events.
   - ✓ Low computation and rewriting overhead.

RWNDQ Algorithm

Switch side (Local window proportional to queue occupancy):

- Increase receive window when below the target.
- Decrease when we are above the queue target.
- Slow start to initially reach target fast.

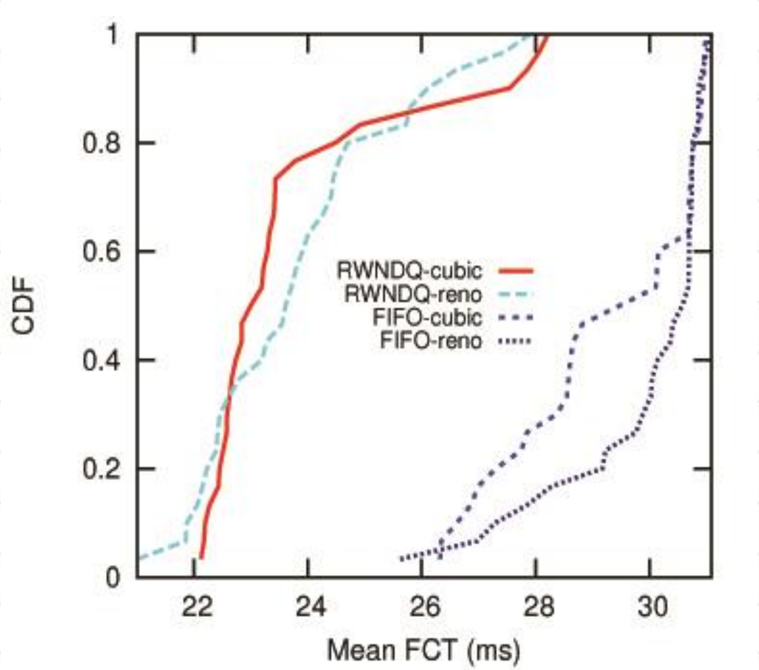


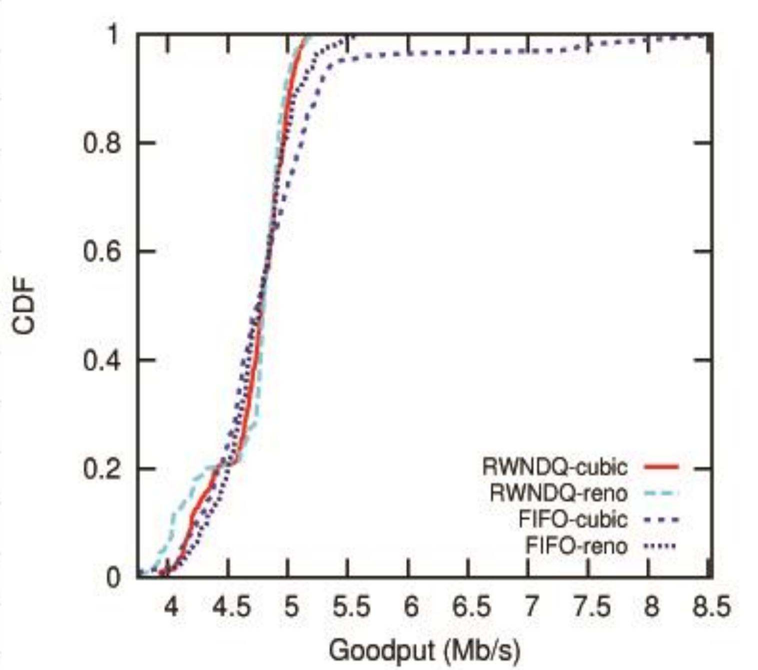
Sender and Receiver side (No Change):

Send Window = Min(Congestion Win, Receive Win)

Sample - Experimental Analysis

- Small Scale Testbed using Open vSwitch
- Scenario depicting 200 elephants against 30 Mice.
- Mice Goal: Low Latency and low variance.
- Elephants Goal: High and enough throughput





Conclusion

- DCN congestion is a hot research topic
  - Business needs and service agreements
  - Quality of service (QoS)
- DCN congestion control is a necessity
  - Incast is a very serious and frequent problem.
  - Employing an efficient packet queueing-scheduling to preserve small switch buffers
  - Meeting deadlines either by achieving low latency or building a deadline-aware networking architecture.

Future Research Directions

- Leveraging functionalities of SDN
- Stability analysis and study.
- Handling persistent TCP connections.
- Adapting to varying initial congestion window.
- Bandwidth allocation in Multi-tenant datacenter with QoS constraints.

THANKS!

QUESTIONS ARE WELCOMED

References

1. V. Jacobson. Congestion avoidance and control. ACM SIGCOMM Computer Communication Review, 18:314-329, 1988.
2. J Dean and S Ghemawat. MapReduce : Simplied Data Processing on Large Clusters. Communications of the ACM, 51:1-13, 2008.
3. Vijay Vasudevan, Amar Phanishayee, Hiral Shah, Elie Krevat, David G. Andersen, Gregory R. Ganger, Garth A. Gibson, and Brian Mueller. Safe and effective fine-grained TCP retransmissions for datacenter communication. ACM SIGCOMM Computer Communication Review, 39:303, 2009.
4. Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. Data center TCP (DCTCP). ACM SIGCOMM Computer Communication Review, 40:63, 2010.
5. Christo Wilson, Hitesh Ballani, Thomas Karagiannis, and Ant Rowstron. Better Never than Late: Meeting Deadlines in Datacenter Networks. In Proc. ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM'11), pages 50{61, 2011.
6. Theophilus Benson, Aditya Akella, and David a. Maltz. Network traffic characteristics of data centers in the wild. In Proceedings of the 10th ACM SIGCOMM page 267, 2010.
7. Mohammad Alizadeh, Shuang Yang, Sachin Katti, Nick McKeown, Balaji Prabhakar, and Scott Shenker. Deconstructing datacenter packet transport. Proceedings of the 11th ACM Workshop on Hot Topics in Networks - HotNets-XI, pages 133-138, 2012.
8. Balajee Vamanan, Jahangir Hasan, and T.N. Vijaykumar. Deadline-aware data center tcp (d2tcp). In Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '12, pages 115{126, New York, NY, USA, 2012. ACM.

References

8. Chi-Yao Hong, Matthew Caesar, and P. Brighten Godfrey. Finishing flows quickly with preemptive scheduling. In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication - SIGCOMM '12, page 127, New York, New York, USA, August 2012. ACM Press.
9. Haitao Wu, Zhenqian Feng, Chuanxiong Guo, and Yongguang Zhang. ICTCP: Incast congestion control for TCP in data-center networks. IEEE/ACM Transactions on Networking, 21:345{358, 2013.
10. Wei Bai, Kai Chen, Haitao Wu, Wuwei Lan, and Yangming Zhao. PAC: Taming TCP Incast congestion Using Proactive ACK Control. In IEEE International Conference on Network Protocols, 2014.
11. Peng Cheng, Fengyuan Ren, Ran Shu, and Chuang Lin. Catch the Whole Lot in an Action: Rapid Precise Packet Loss Notication in Data Center. In Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), pages 17{28, 2014.
12. Ahmed M. Abdelmoniem and Brahim Bensaou, “Incast-Aware Switch-Assisted TCP Congestion Control for Data Centers”, IEEE Global Communications Conference (Globecom 15).
13. Ahmed M. Abdelmoniem and Brahim Bensaou, “Reconciling Mice and Elephants in Data Center Networks”, IEEE Cloud Networking Conference (Cloudnet 15).
14. Ahmed M. Abdelmoniem and Brahim Bensaou, “ Efficient Switch-Assisted Congestion Control for Data Centers: an Implementation and Evaluation”, IEEE Performance Computing and Communications Conference (IPCCC15).