

Towards High-Availability for IP Telephony using Virtual Machines

Authors : Patnaik, D.; Bijlani, A.; Singh, V.K.;

Source : Internet Multimedia Services Architecture
and Application(IMSAA), 2010 IEEE 4th
International Conference on

Advisor : Whai-En Chen

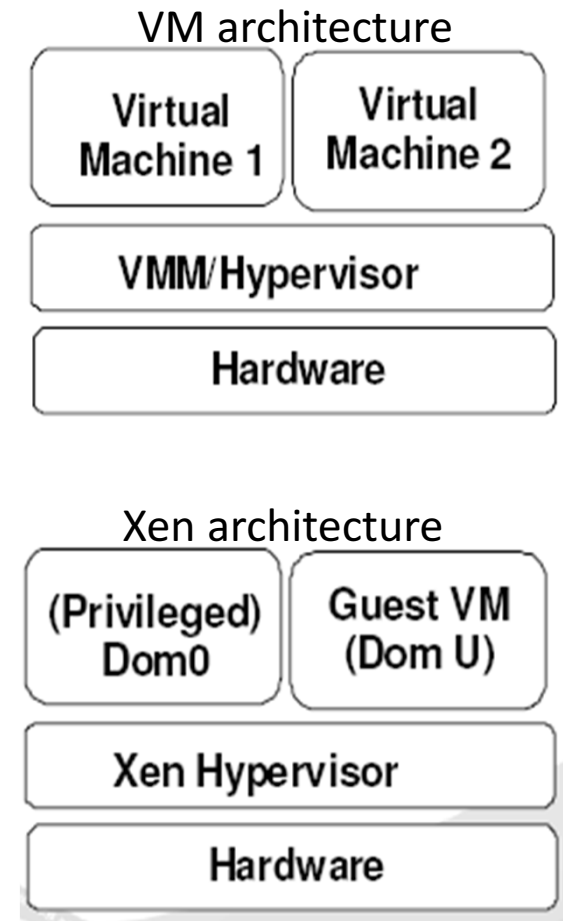
Student : Jun-Jie CIOU

Outline

- Virtualization and its Benefit
- Virtualization and High Availability (HA) benefits
- Live migration
- Remus on Xen
- Reliability and Scalability using Virtual Machines
- Analysis and results :
 - Signaling
 - Media
- Conclusion

Virtualization and its Benefit

- Abstraction layer (Hypervisor) between the physical hardware and the OS.
- Single physical machine can host multiple virtual machines each running a different OS + application stack
- VMMs (Virtual Machine Manager)
 - Xen, VMWare, VirtualBox, Microsoft HyperV
- Benefits
 - Server consolidation
 - Green computing
 - Cost savings – space and power
 - **High Availability**



Dom : Domain

VM : Virtual Machine

Virtualization and High Availability (HA) benefits

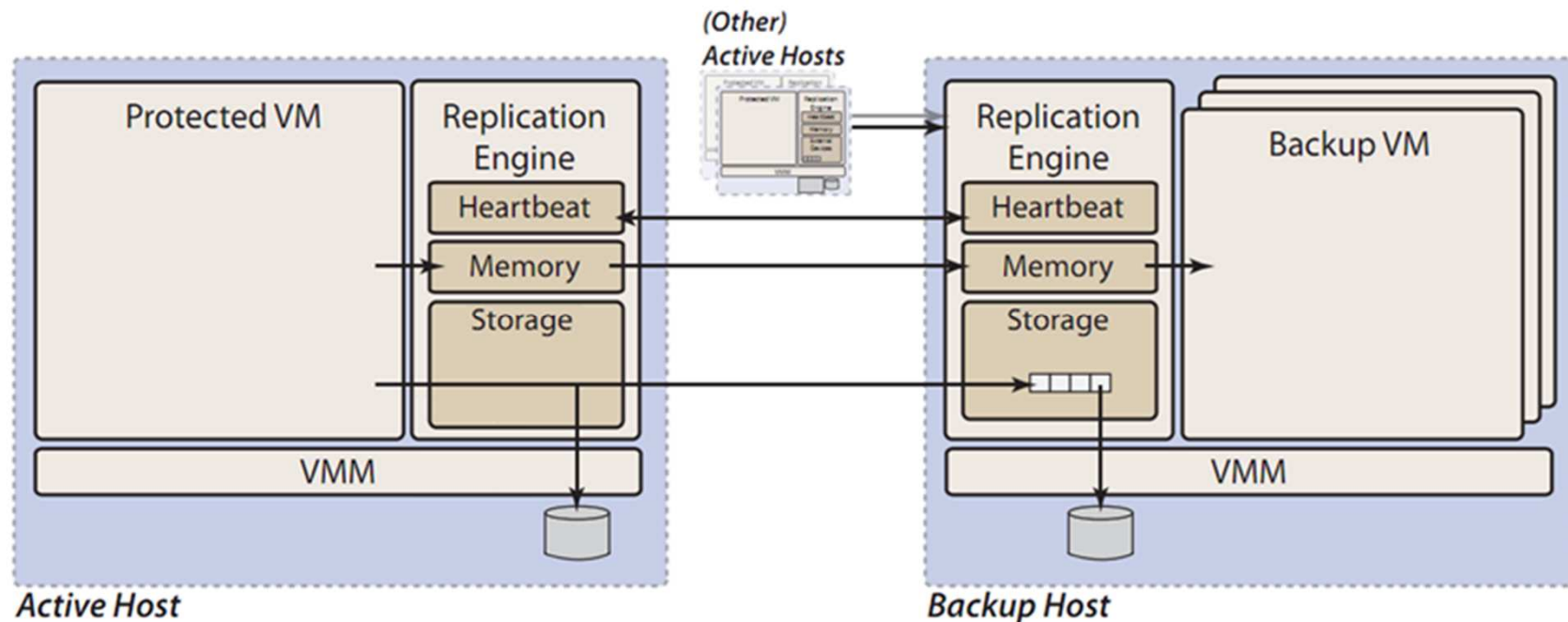
- Seamless fail-over, Efficient and transparent migration of VM to another physical machine
 - Minimal or no impact to client nodes
 - Continuously syncs the state between the primary and secondary host
- This paper use
 - [Remus: A High Availability Solution for Xen](#)

Live migration

- Live migration that supports migration of actively executing VMs from one physical machine to another with minimal disruption in service.
- This capability allows administrators to provide high-availability.
- Moreover, software applications do not need to implement failover and recovery logic.

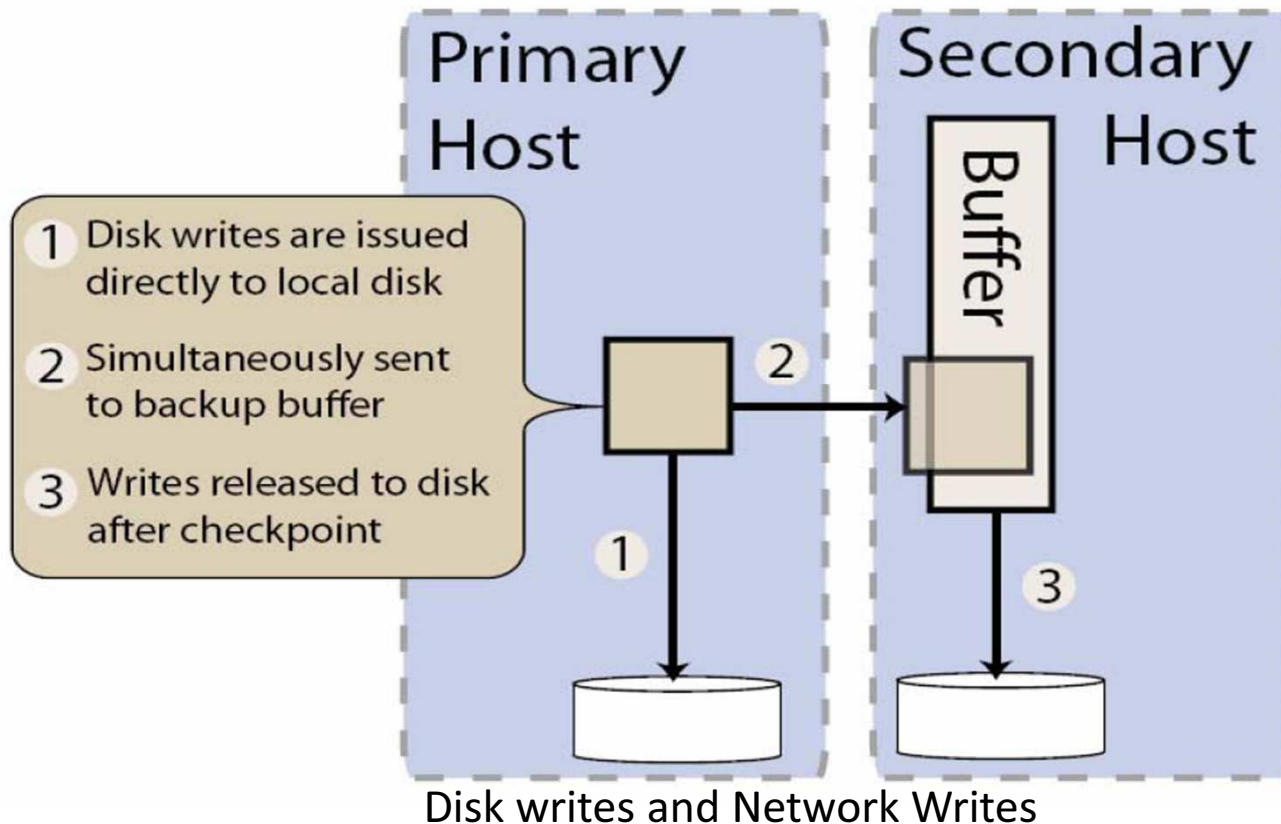
Remus on Xen (1/2)

- Remus uses continuous check-pointing and keeps a consistent client view of network state
- The secondary machine hosts a paused replica of the primary VM
- Uses a heartbeat mechanism
 - Failure to receive periodic heartbeat on secondary will un-pause the backup VM
 - Heartbeat time-out can be configured



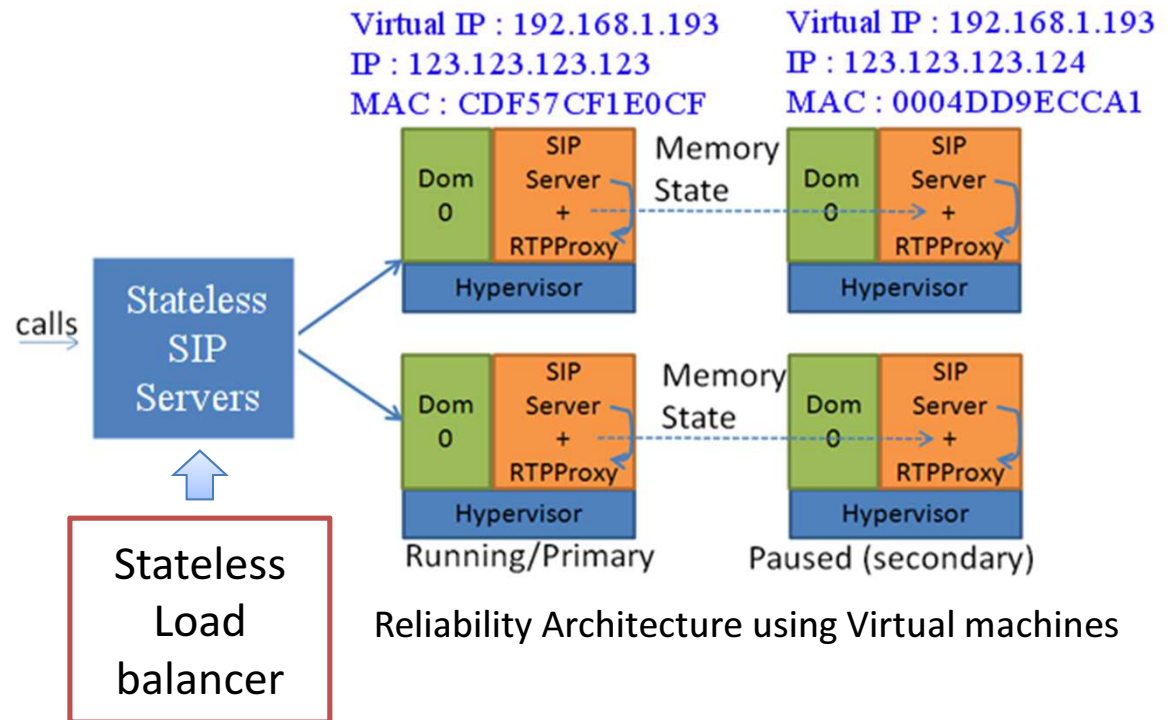
Remus on Xen (2/2)

- Remus modes of operation
 - Net Mode : Highly reliable
 - No-Net Mode : better performance with negligible packet loss in case of failure



Reliability and Scalability using Virtual Machines

- Scalability using load balancer (LB)
 - LB can elastically add more VMs as demand grows



Gratuitous ARP

- When primary heartbeat is missed, The backup VM is active, and using an unsolicited ARP reply (Gratuitous ARP), the network traffic from clients is redirected to it. This ensures seamless fail-over.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 16 arrived at 11:48:14.4769; frame size is 60 (003C hex) bytes.
DLC: Destination = BROADCAST FFFFFFFF, Broadcast
DLC: Source = Station Cisco 9ECCA1
DLC: Ethertype = 0806 (ARP)
DLC:
ARP: ----- ARP/RARP frame -----
ARP:
ARP: Hardware type = 1 (10Mb Ethernet)
ARP: Protocol type = 0800 (IP)
ARP: Length of hardware address = 6 bytes
ARP: Length of protocol address = 4 bytes
ARP: Opcode 2 (ARP reply)
ARP: Sender's hardware address = 0004DD9ECCA1
ARP: Sender's protocol address = [192.168.1.193]
ARP: Target hardware address = FFFFFFFF
ARP: Target protocol address = [192.168.1.193]
ARP:
ARP:
DLC: Frame padding= 18 bytes

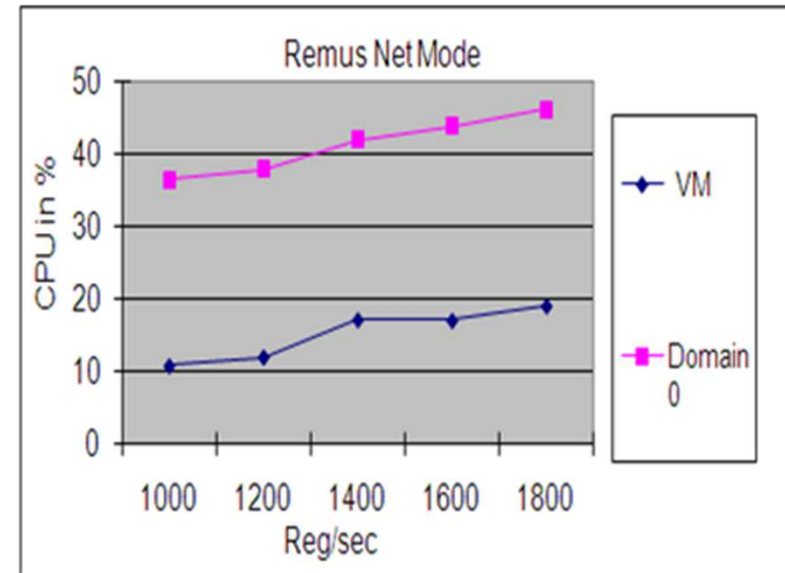
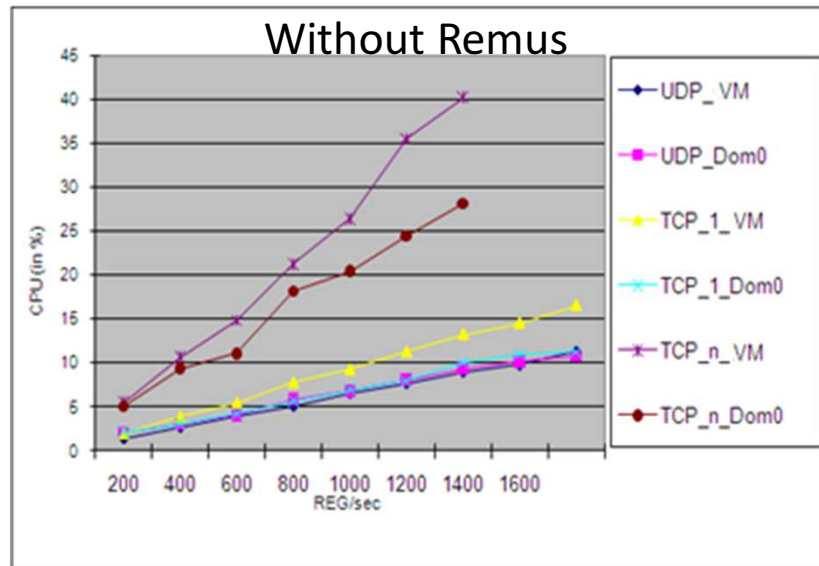
00000000: ff ff ff ff ff ff 00 04 dd 9e cc a1 08 06 00 01 .. 00..
00000010: 08 00 06 04 00 02 00 04 dd 9e cc a1 c0 a8 01 c1 ..... 00..
00000020: ff ff ff ff ff ff c0 a8 01 c1 00 00 00 00 00 00 .. 00..
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .. 00..
```

Studying performance Implications

- Experimental setup
 - Primary /Backup Servers
 - Intel Core 2 Quad Processors, 2.5 Ghz, 8 GB RAM, 4MB L2 Cache
 - Hypervisor – Xen 3.2.1 + Remus
 - Guest OS : Para Virtualized Linux 2.6.18
- IP Telephony Workload
 - Modeled our workload using [SIPstone](#)
 - Measured % success of registrations during failover
 - Used UDP and TCP as transport for registrations
 - Used [OpenSIPs](#) as SIP server
 - [RTPProxy](#) as Media Server
 - [SIPp](#) for generating signaling and media traffic

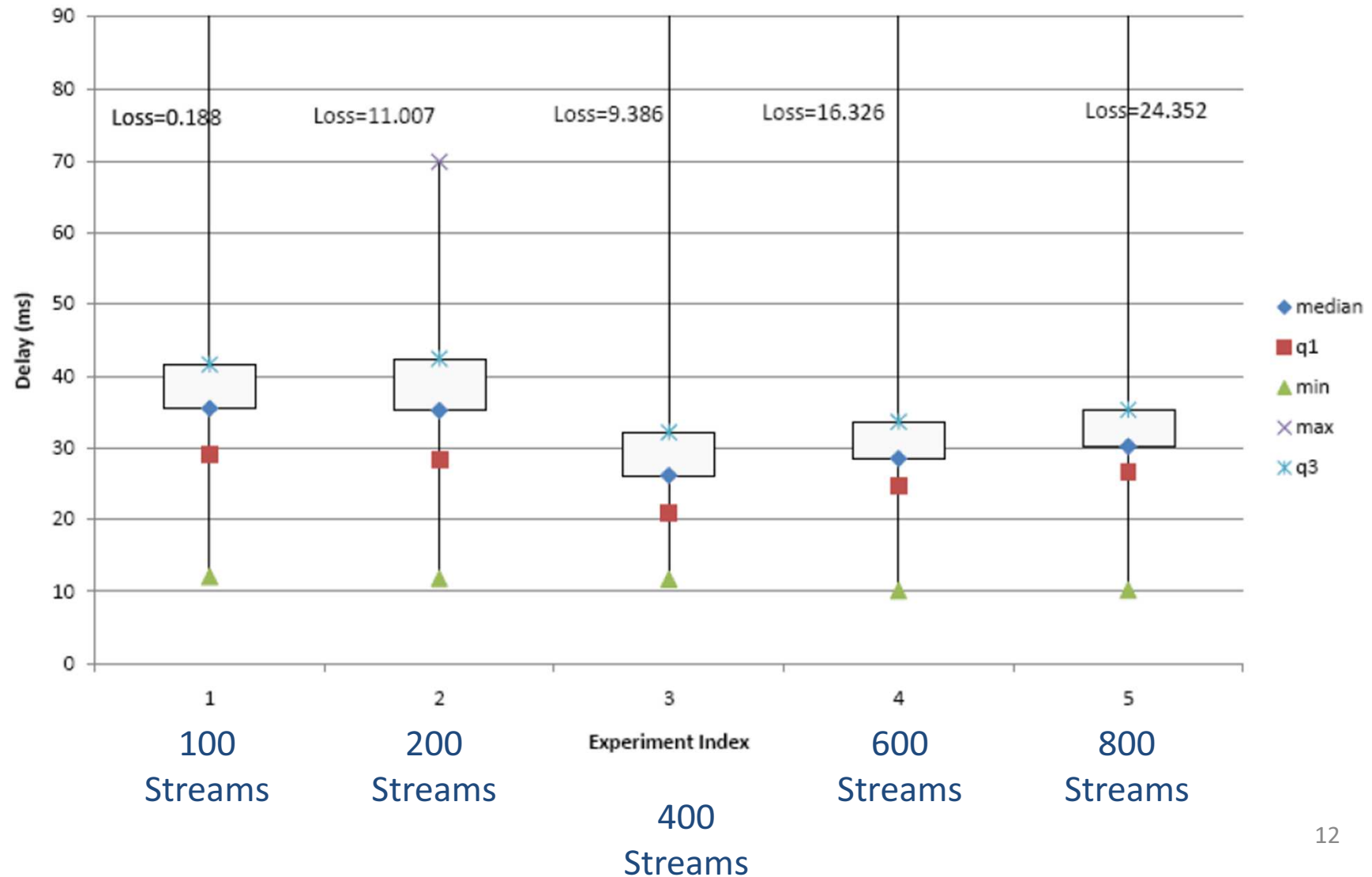
Analysis and results : signaling

- VM and Domain 0 both have high CPU utilization with TCP_N (new tcp connection for each REGISTER)
- UDP and TCP_1 (1 tcp connection for all REGISTER) have similar overhead.



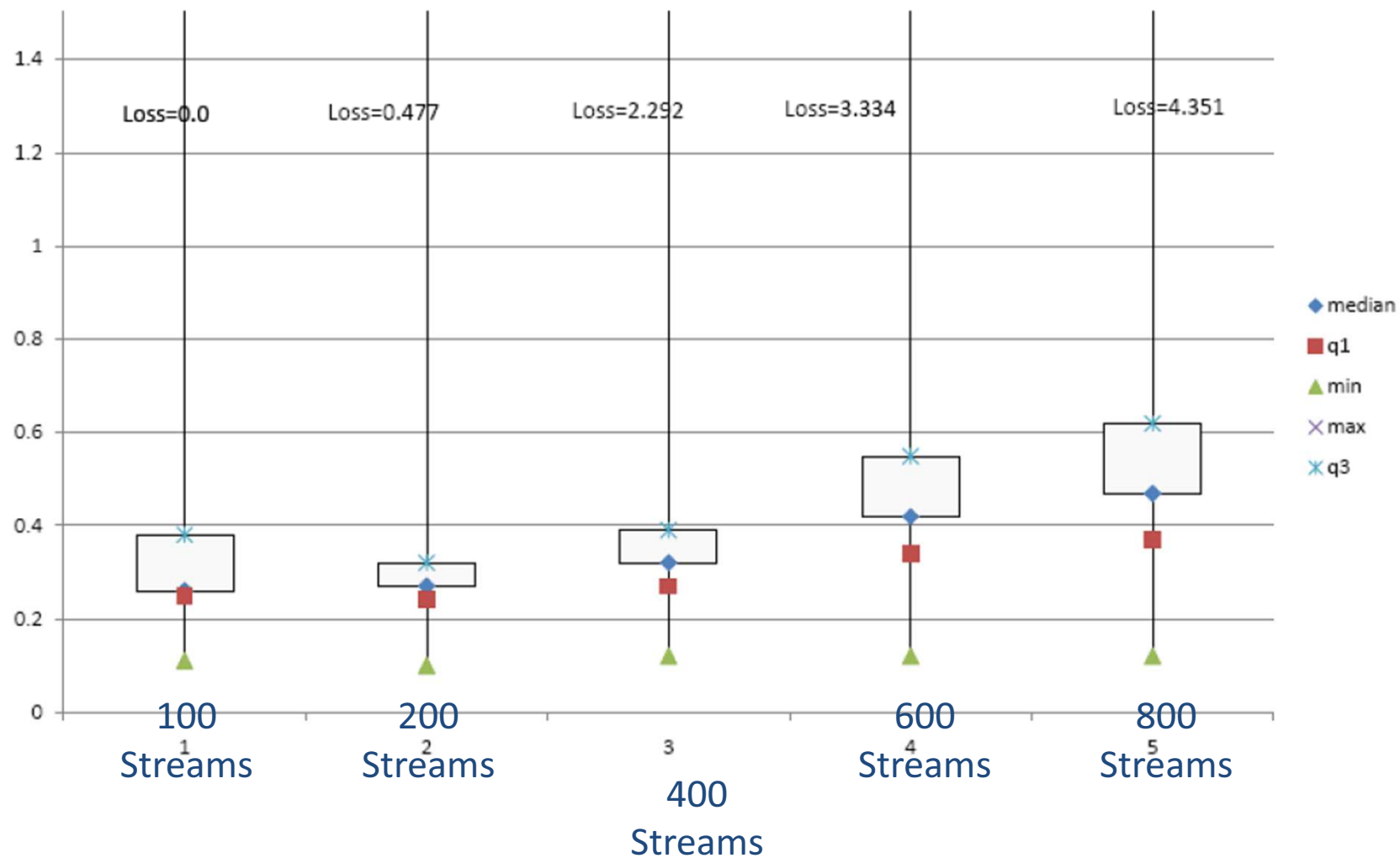
CPU utilization (in VM, Dom0)
UDP means with UDP transport,
TCP_1 means same connection for all call,
TCP_N means new connection for each call

Analysis and results: media Net mode



Analysis and Results: Media

No-net mode



Conclusion

- **Using No-Net mode for media streams gives us a balance between performance(loss and delay) and reliability(failover) while still being able to migrate 100% of all calls in progress (using TCP) which is a significant result**
- Net Mode for Signaling is a good configuration with 100% registration completion with failover
- No-Net mode for the Media server deployment provides significant improvement in performance: loss and delay reduces significantly