LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MNM TEAM
MUNICH NETWORK MANAGEMENT TEAM

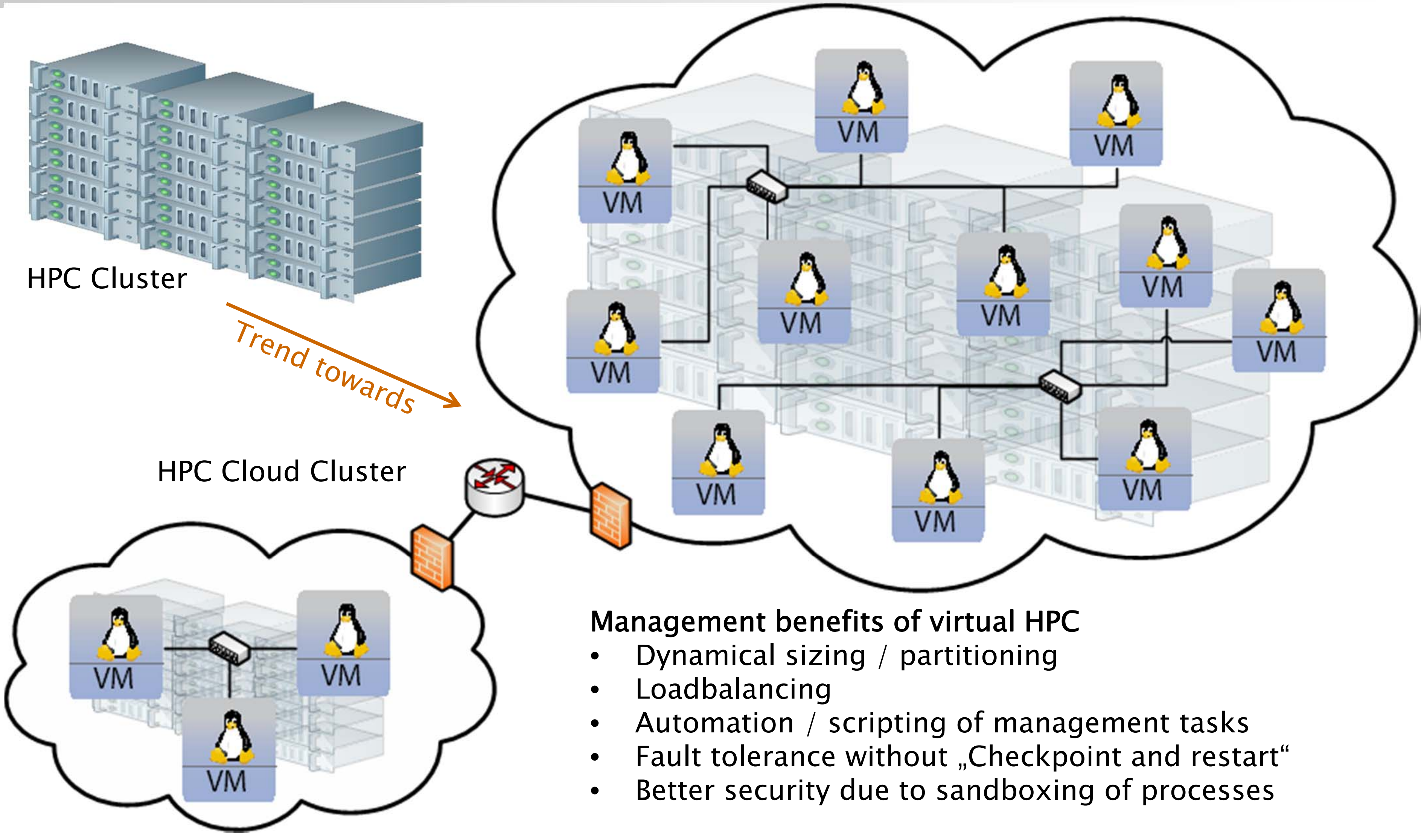
lrz

SVM 2010

High-performance aspects in virtualized infrastructures

Vitalian Danciu, Nils gentschen Felde,
Dieter Kranzlmüller, Tobias Lindinger

Motivation – Idea



Management benefits of virtual HPC

- Dynamical sizing / partitioning
- Loadbalancing
- Automation / scripting of management tasks
- Fault tolerance without „Checkpoint and restart“
- Better security due to sandboxing of processes

Motivation – Challanges

MNM TEAM

But: What about Performance?

- Effects expected:
  - –Virtualization layer will induce overhead
  - –Concurrent effects appear when running several VMs in parallel; e.g. effects on storage, network, memory access
  - –Effects will be dependent on virtualization architecture, implementations, Operating Systems, …
- What scale will that effects be?
- How to measure?

Outline

High-performance aspects in virtualized infrastructures

Vitalian Danciu, Nils gentschen Felde, Dieter Kranzlmüller, Tobias Lindinger

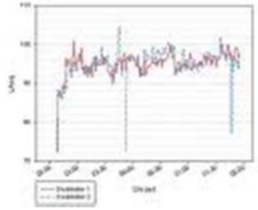
Content:

- –Designing a Benchmark Suite
- –Applying the Benchmark Suite
- –Impacts to HEC Problem Classes
- –Conclusion
- –Outlook & Further research

Benchmark Suite – Design Goals

- Measure core components of virtualized systems
  - CPU
  - RAM
  - Network
  - Disk

- Compare results of different VMM
  - architectures (full-, native-, … , paravirtualization)
  - Implementations (VMware ESXi, Xen, MS Hyper-V, …)
- Analyze impact of virtualization on different
  - Operating systems (windows, linux, …)
  - System architectures (32 bit, 64 bit, …)
- Measure
  - Overhead of virtualization
  - Concurrent effects

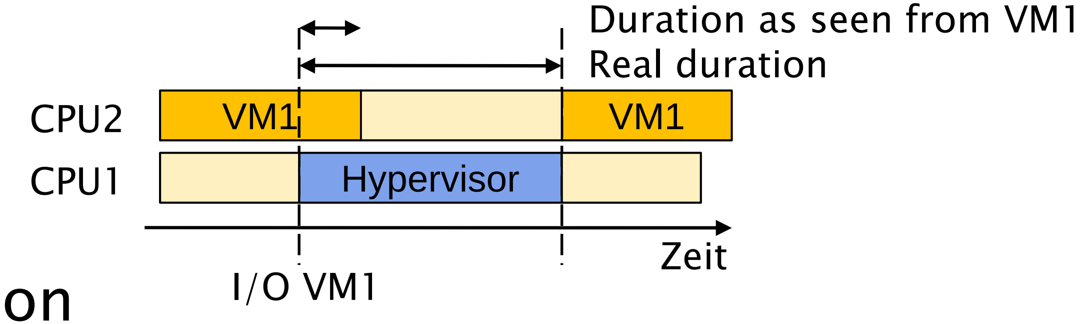


- Take care about special time measurement in VMs

Benchmark Suite – Time Measurement

Measurement of time is difficult in virtualized environments:

- –Definition of time
- –Time synchronisation



Example: Linpack-Benchmark

```
static REAL second(void){
    return ((REAL)((REAL)clock()/(REAL)CLOCKS_PER_SEC));
}
# clock()        : Wall-Clock-Time in time ticks since process start
# CLOCKS_PER_SEC: Frequency of the hardware clock
```

→ External clocks may be necessary, depending on counters used by the benchmark tool

Benchmark Suite – Implementation

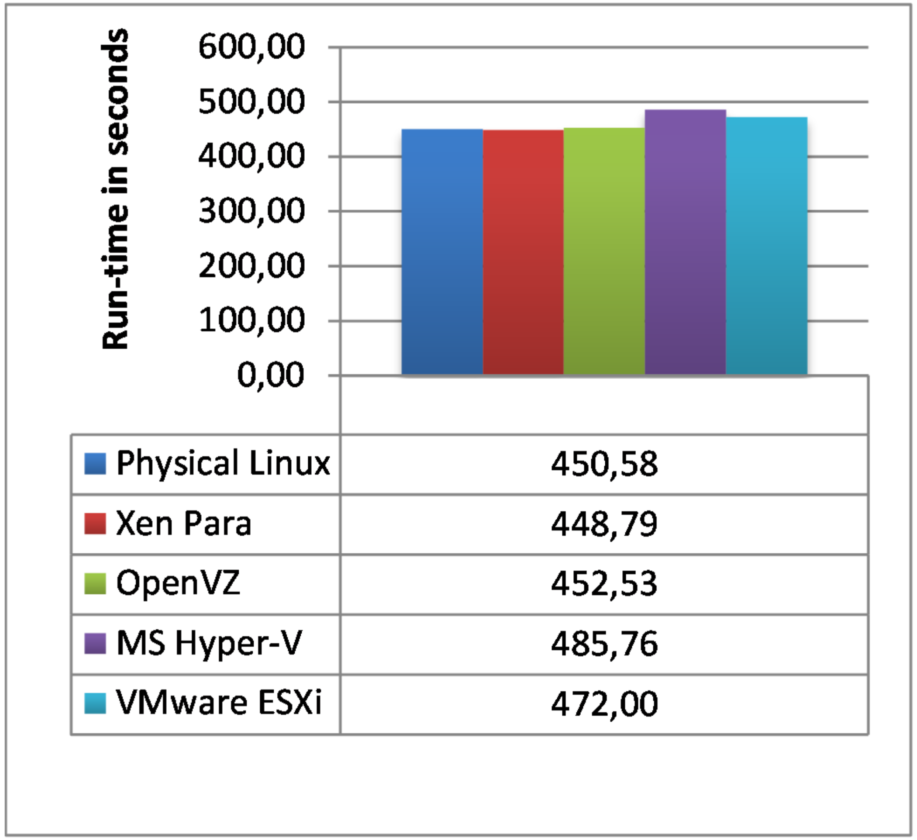
- Benchmarks used:

Component	Benchmark
CPU	Linpack
RAM	Ramspeed
Disk	Iometer
Network	Iometer

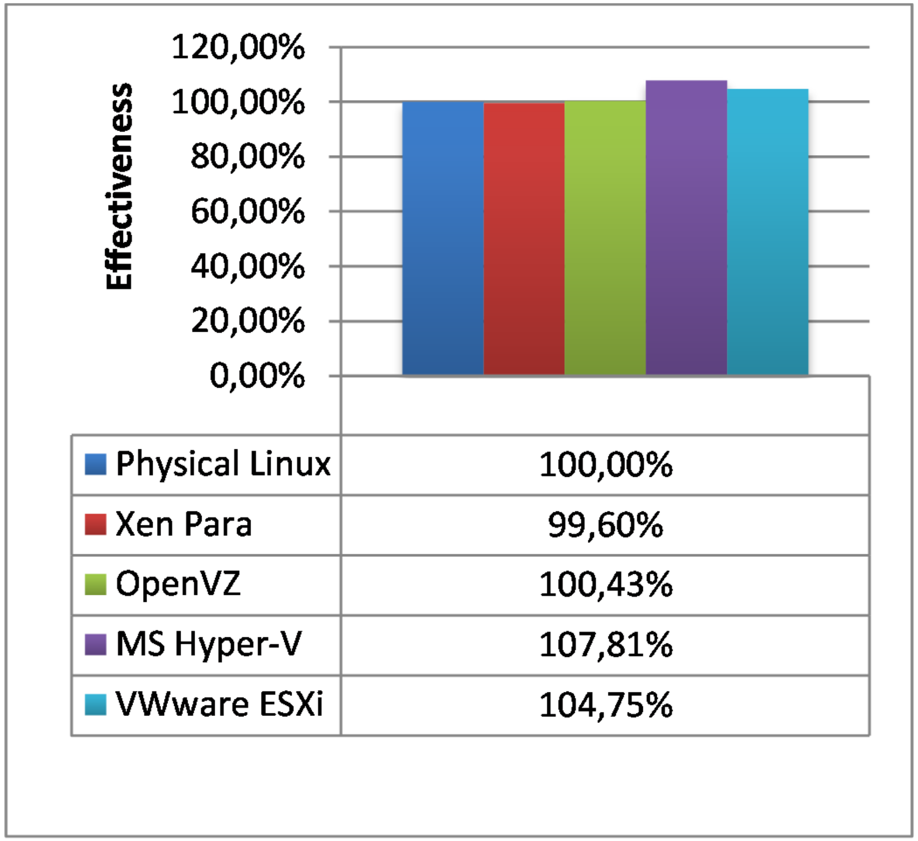
- Run Combinations of single benchmarks to test for concurrent effects, e.g.
  - CPU + Network
  - Parallel network access
  - Different network setups (VM2VM, VM2PM, …)
  - …

Applying the Test Suite

Linpack – Absolute/Relative Runtime



Physical Linux	450,58
Xen Para	448,79
OpenVZ	452,53
MS Hyper-V	485,76
VMware ESXi	472,00



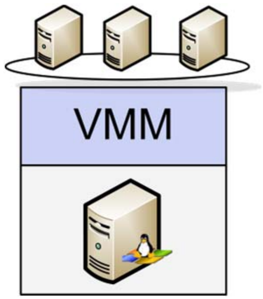
Physical Linux	100,00%
Xen Para	99,60%
OpenVZ	100,43%
MS Hyper-V	107,81%
VWware ESXi	104,75%

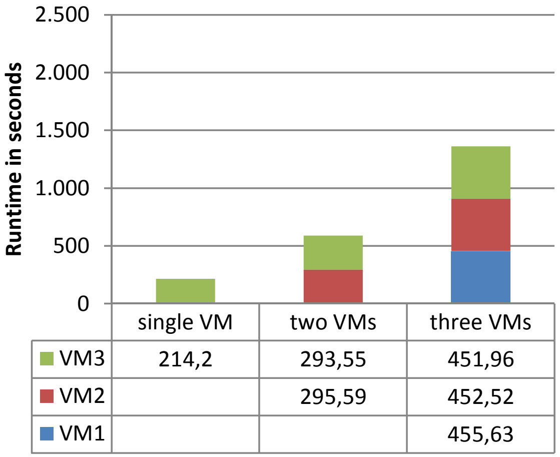
Absolute Runtime

Relative Runtime

Applying the Test Suite

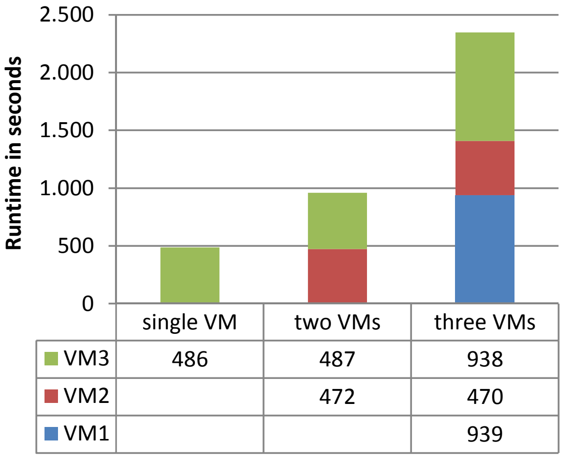
CPU – Concurrent effects





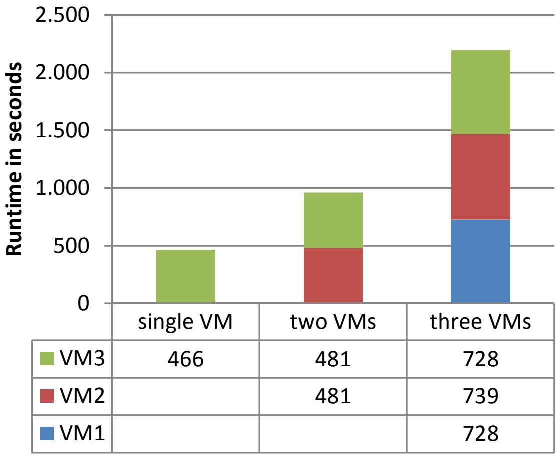
	single VM	two VMs	three VMs
🟩 VM3	214,2	293,55	451,96
🟥 VM2		295,59	452,52
🟦 VM1			455,63

Xen



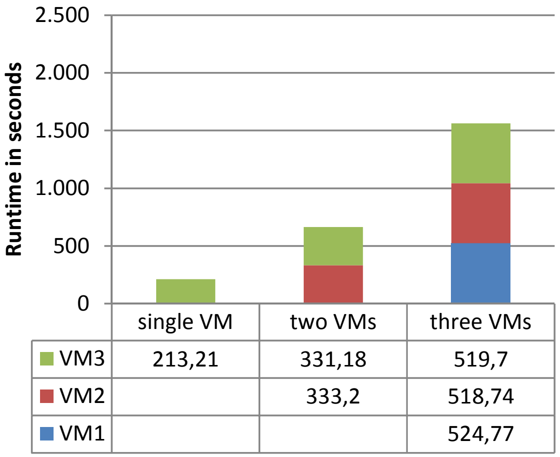
	single VM	two VMs	three VMs
🟩 VM3	486	487	938
🟥 VM2		472	470
🟦 VM1			939

Virtuozzo



	single VM	two VMs	three VMs
🟩 VM3	466	481	728
🟥 VM2		481	739
🟦 VM1			728

Hyper-V

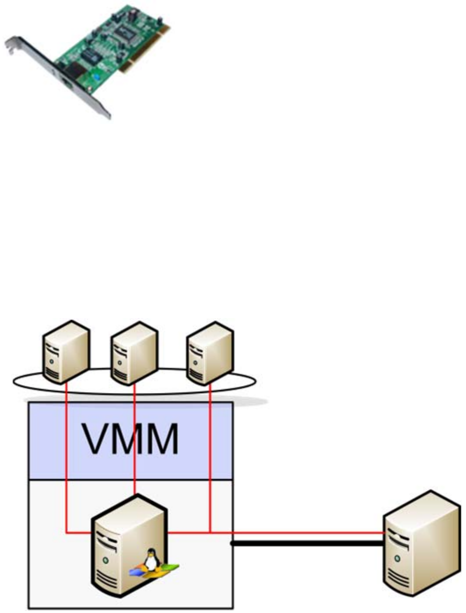


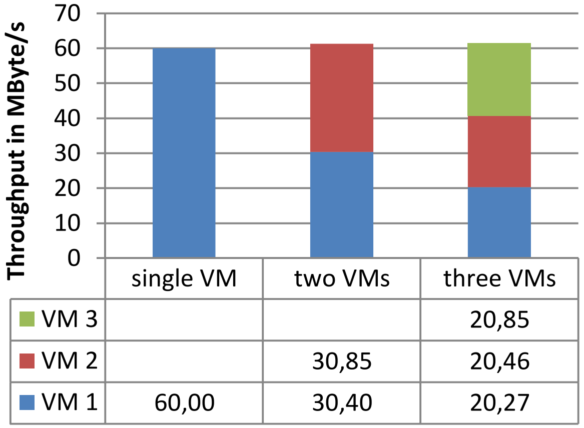
	single VM	two VMs	three VMs
🟩 VM3	213,21	331,18	519,7
🟥 VM2		333,2	518,74
🟦 VM1			524,77

ESXi

Applying the Test Suite

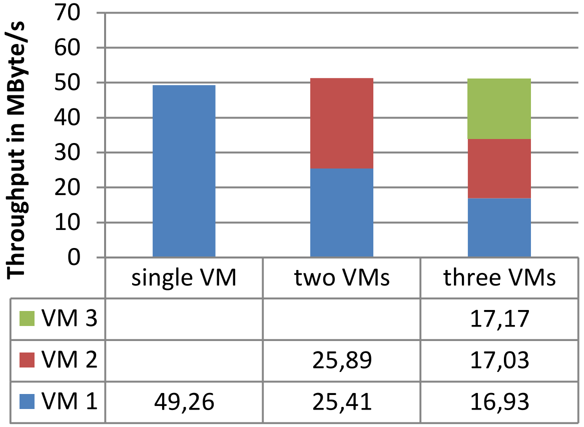
Network – Concurrent effects





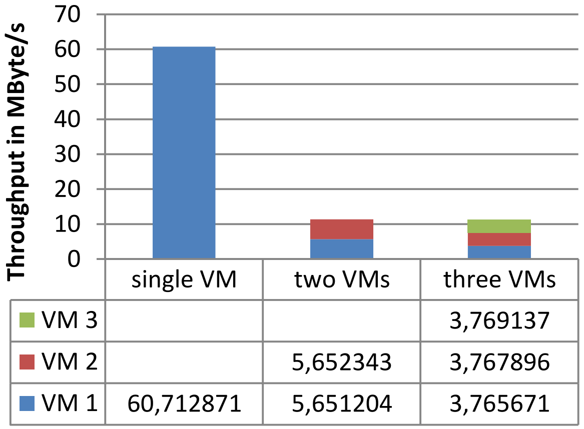
	single VM	two VMs	three VMs
🟩 VM 3			20,85
🟥 VM 2		30,85	20,46
🟦 VM 1	60,00	30,40	20,27

Xen – receive data from network



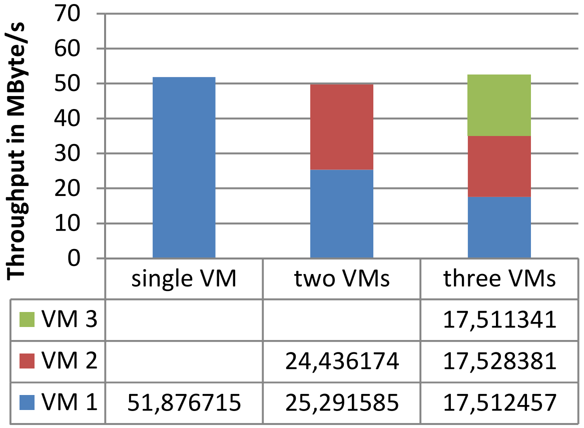
	single VM	two VMs	three VMs
🟩 VM 3			17,17
🟥 VM 2		25,89	17,03
🟦 VM 1	49,26	25,41	16,93

Xen – send data to network



	single VM	two VMs	three VMs
🟩 VM 3			3,769137
🟥 VM 2		5,652343	3,767896
🟦 VM 1	60,712871	5,651204	3,765671

Virtuozzo – receive data from network

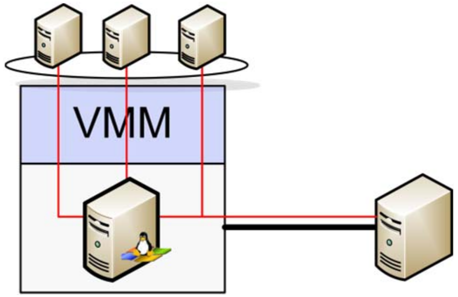


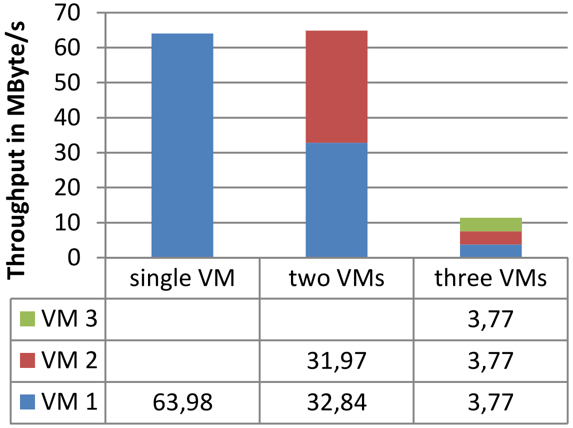
	single VM	two VMs	three VMs
🟩 VM 3			17,511341
🟥 VM 2		24,436174	17,528381
🟦 VM 1	51,876715	25,291585	17,512457

Virtuozzo – send data to network

Applying the Test Suite

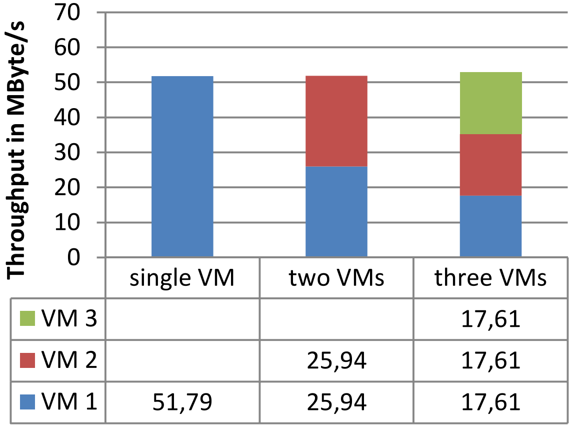
Network – Concurrent effects





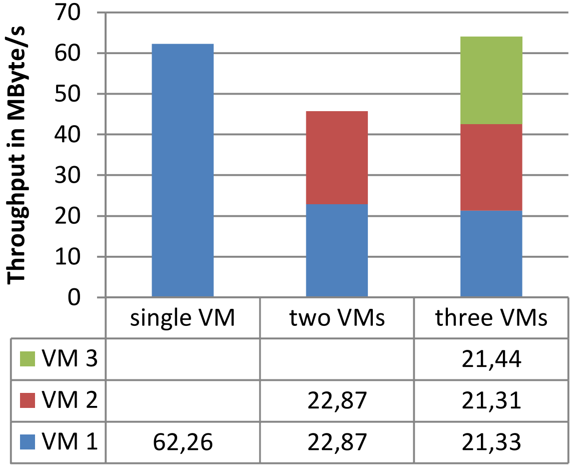
	single VM	two VMs	three VMs
🟩 VM 3			3,77
🟥 VM 2		31,97	3,77
🟦 VM 1	63,98	32,84	3,77

Hyper-V – receive data from network



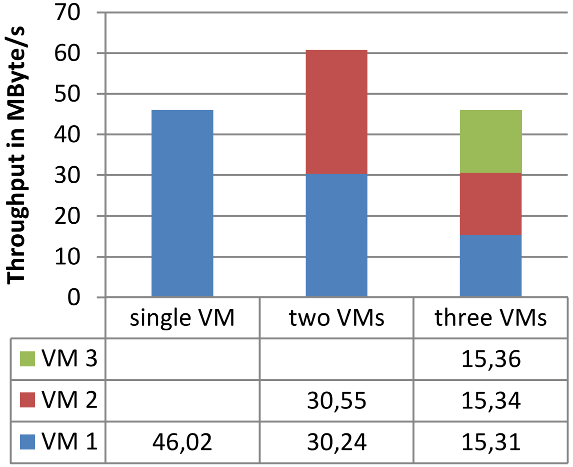
	single VM	two VMs	three VMs
🟩 VM 3			17,61
🟥 VM 2		25,94	17,61
🟦 VM 1	51,79	25,94	17,61

Hyper-V– send data to network



	single VM	two VMs	three VMs
🟩 VM 3			21,44
🟥 VM 2		22,87	21,31
🟦 VM 1	62,26	22,87	21,33

ESXi– receive data from network

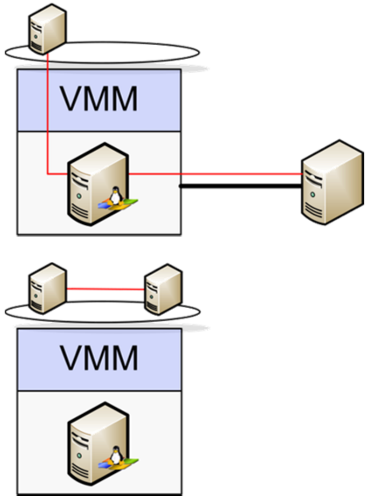


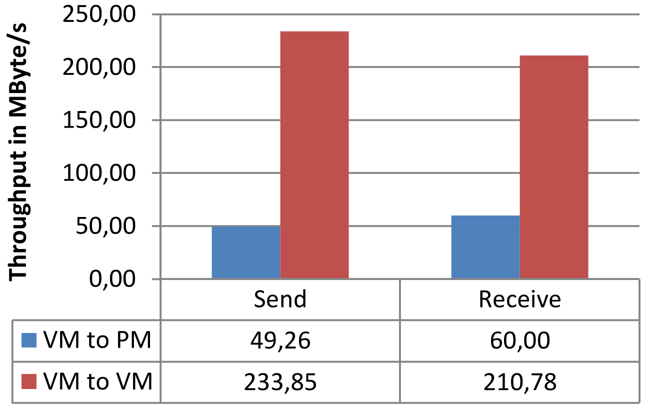
	single VM	two VMs	three VMs
🟩 VM 3			15,36
🟥 VM 2		30,55	15,34
🟦 VM 1	46,02	30,24	15,31

ESXi– send data to network

Applying the Test Suite

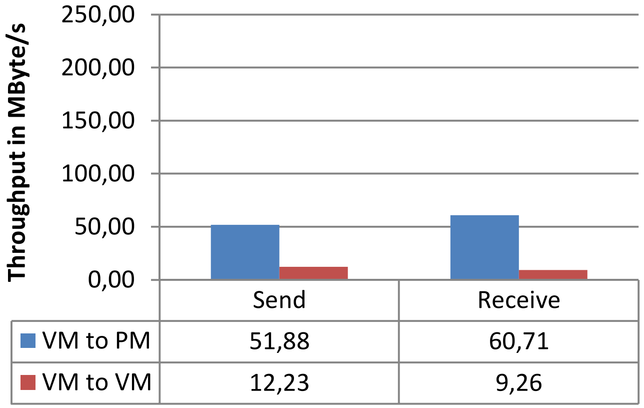
Network – Physical vs. virtual communication peers





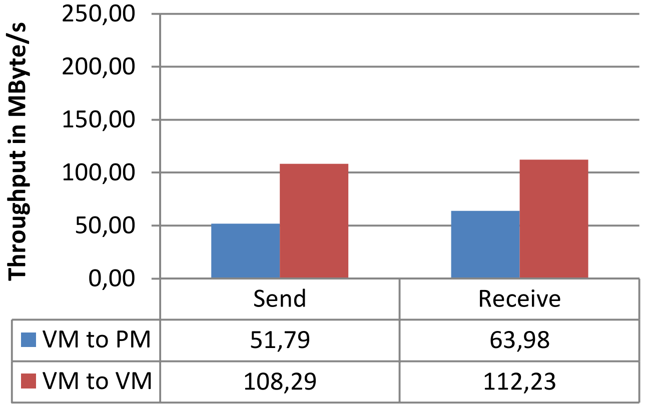
	Send	Receive
🟦 VM to PM	49,26	60,00
🟥 VM to VM	233,85	210,78

Xen



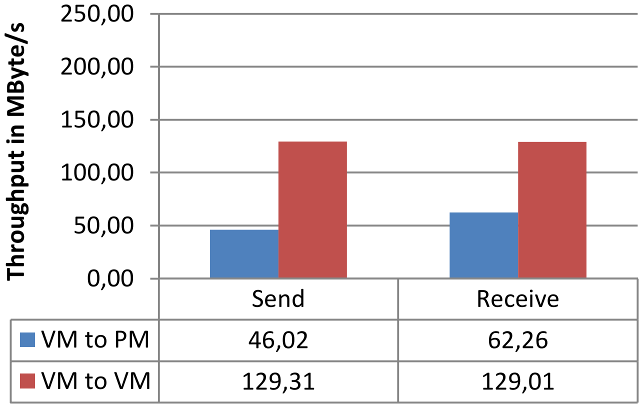
	Send	Receive
🟦 VM to PM	51,88	60,71
🟥 VM to VM	12,23	9,26

Virtuozzo



	Send	Receive
🟦 VM to PM	51,79	63,98
🟥 VM to VM	108,29	112,23

Hyper-V



	Send	Receive
🟦 VM to PM	46,02	62,26
🟥 VM to VM	129,31	129,01

ESXi

Performance Reqirements

Criterion	High-Performance	High-Throughput
Coupling	tight	loose
CPU impact	co-determinant	critical
Interconnect impact	critical	less
RAM addressing	sensitive; prefers contiguous, uniform latency addressing	less sensitive
Program structures	inter-communicating program replicas	workflows; pipelining of computing tasks
Examples	fluid dynamics problems, crash codes	high-energy physics (e.g. CERN LHC experiments), general parameter variation studies

Conclusion

- CPU virtualization is efficient (close to 100%)
- RAM access is fast

→ High Througput Computing (HTC) Tasks can be virtualized efficiently

- Network Access efficiency depends on
  - flow direction
  - concurrent use
  - Available CPU power
- Topology changes („vMotion“) confuses MPI

→ High Performance Computing (HPC) Tasks should not be virtualized yet

Outlook

- Technical realization of virtual High Throughput Computing
- More Research for virtualized High Performance Computing needed:
  - Adapt MPI to detect topology changes caused by live migrations
  - Calculate distributions of VMs with minimal communication overhead
  - Maybe allocate physical NICs to dedicated VM
  - Use additional I/O virtualization approaches

Questions?