TPC-V: A Benchmark for Evaluating the Performance of Database Applications in Virtual Environments

Priya Sethuraman and H. Reza Taheri, VMWare, Inc.
{psethuraman, rtaheri}@vmware.com

TPCTC 2010
Singapore

vmware

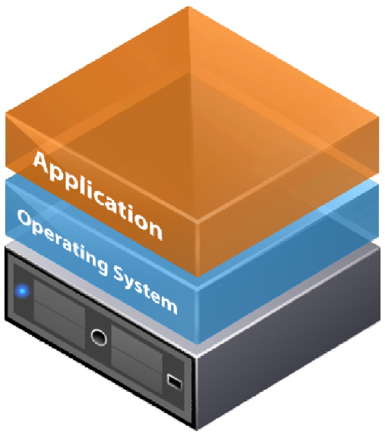
Agenda/Topics

- Introduction to virtualization
- Existing benchmarks
- Genesis of TPC-V
- But what is *TPC-E*???
- TPC-V design considerations
- Set architectures, variability, and elasticity
- Benchmark development status
- Answers to some common questions

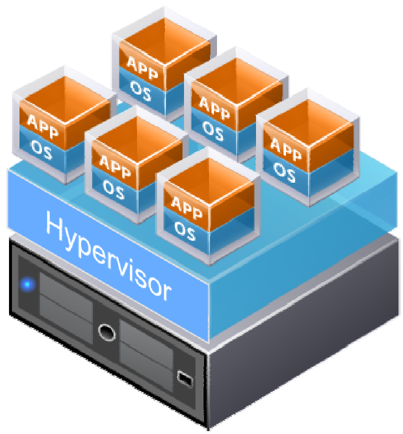
What is a Virtual Machine?

A (VM) is a software computer that, like a physical computer, runs an operating system and applications. An operating system installed on a virtual machine is called a guest operating system. Virtual machines run on host servers. The same server can run many virtual machines. Every VM runs in an isolated environment.

- **Started out with IBM VM in the 60s**
- **Also on Sun Solaris, HP Itanium, IBM Power/AIX, others**
- **A new wave started in the late 90s on X86**
  - Initially, enthusiasts ran Windows and Linux VMs on their PCs



Traditional Architecture



Virtual Architecture

Why virtualize a *server*?

- **Server consolidation**
  - The vast majority of server are grossly underutilized
  - Reduces both CapEx and OpEx
- **Migration of VMs (both storage and CPU/memory)**
  - Enables live load balancing
  - Facilitates maintenance
- **High availability**
  - Allows a small number of generic servers to back up all servers
- **Fault tolerance**
  - Lock-step execution of two VMs
- **Cloud computing! Utility computing was finally enabled by**
  - Ability to consolidate many VMs on a server
  - Ability to live migrate VMs in reaction to workload change

Databases: Why Use VMs for databases?

- **Virtualization at hypervisor level provides the best abstraction**
  - Each DBA has their own hardened, isolated, managed sandbox
- **Strong Isolation**
  - Security
  - Performance/Resources
  - Configuration
  - Fault Isolation
- **Scalable Performance**
  - Low-overhead virtual Database performance
  - Efficiently stack databases on one host

Need for a benchmark

- Virtualization is becoming the norm for servers, especially database servers
- Server buyers rely on benchmarks to make purchasing decisions
- Benchmarking database servers is complex; it requires a well crafted specification
- Don't want to go back to the Wild West days of the 1980s benchmark wars

It follows that we need an industry standard benchmark with a DBMS workload

Today’s visualization benchmarks

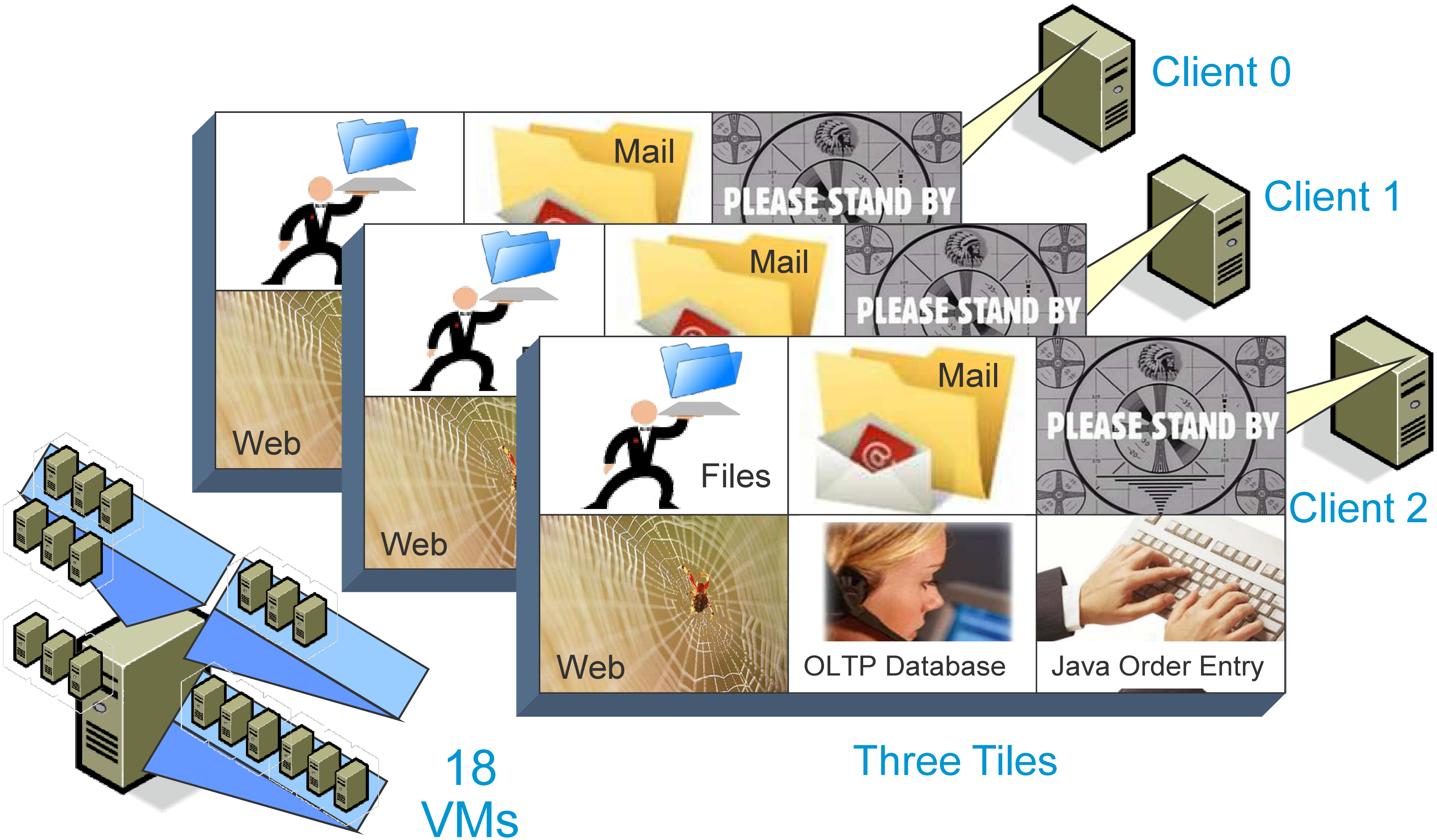
- **VMmark**
  - Developed by VMware in 2007
  - De facto industry standard
  - 120 results from 13 vendors
    - TPC-E, which came out at the same time, lists 39 disclosure from 7 vendors
- **SPECvirt**
  - Industry standard
  - Was released this summer
  - But not a DBMS/backend server benchmark
- **vConsolidate**
  - Developed by IBM and Intel in 2007
- **vApus Mark I from Sizing Server Lab**
- **vServCon developed for internal use by Fujitsu Siemens Computers**

VMmark

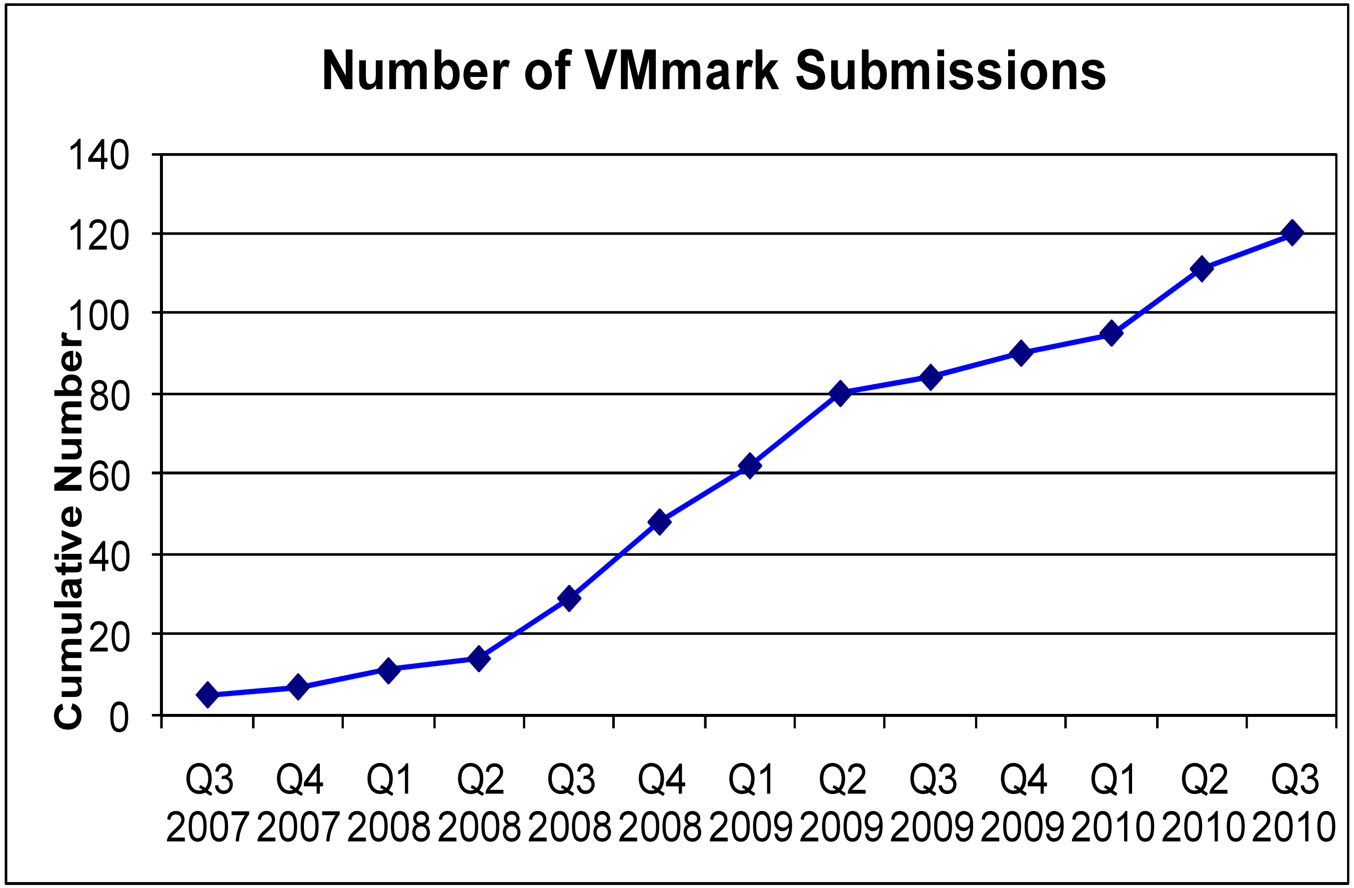
- Aimed at server consolidation market
- ***A mix of workloads***
  - ***Tile*** is a collection of VMs executing a set of diverse workloads

Workload	Application	Virtual Machine Platform
Mail server	Exchange 2003	Windows 2003, 2 CPU, 1GB RAM, 24GB disk
Java server	SPECjbb®2005-based	Windows 2003, 2 CPU, 1GB RAM, 8GB disk
Standby server	None	Windows 2003,1 CPU, 256MB RAM, 4GB disk
Web server	SPECweb®2005-based	SLES 10, 2 CPU, 512MB RAM, 8GB disk
Database server	MySQL	SLES 10, 2 CPU, 2GB RAM, 10GB disk
File server	dbench	SLES 10, 1 CPU, 256MB RAM, 8GB disk

VMmark client workload drivers



VMmark is the de-facto Virtualization Benchmark



(as of 9/2010)

SPECvirt_sc2010

- **Released on 7/14/2010**
  - 2 results published
- **Similar to VMmark**
  - Tile architecture
    - 6 VMs per tile
  - Low disk I/O
    - 96 drives in RAID5 for a 2-socket, 12-core Westmere EP (X5680) server
      - Same caliber systems with *TPC-E* used 532-584 drives in RAID1
- **Different from VMmark**
  - Industry standard benchmark
  - No hypervisor specified
  - Free hand in choosing the software stack

So why do we need a new benchmark?

- The prominent virtual benchmarks today cover consolidation of ***diverse workloads***
- None are aimed at transaction processing or decision support applications, the traditional areas addressed by TPC benchmarks.
- The new frontier is virtualization of resource-intensive workloads, including those which are distributed across multiple physical servers.
- None of the existing virtual benchmarks available today measure the database-centric properties that have made TPC benchmarks the industry standard that they are today.

Birth of a new benchmarks

- **We presented a paper in the 2009 TPCTC**
  - Quantitative data showing that you can virtualize databases
  - But no proper benchmark exists
- **At the December 2009 TPC meeting, the General Council passed the following:**

  Recommend that the GC form a working group to scope a virtualization benchmark that is not comparable to other benchmarks and report back on options to the GC at the February meeting

Benchmark requirements

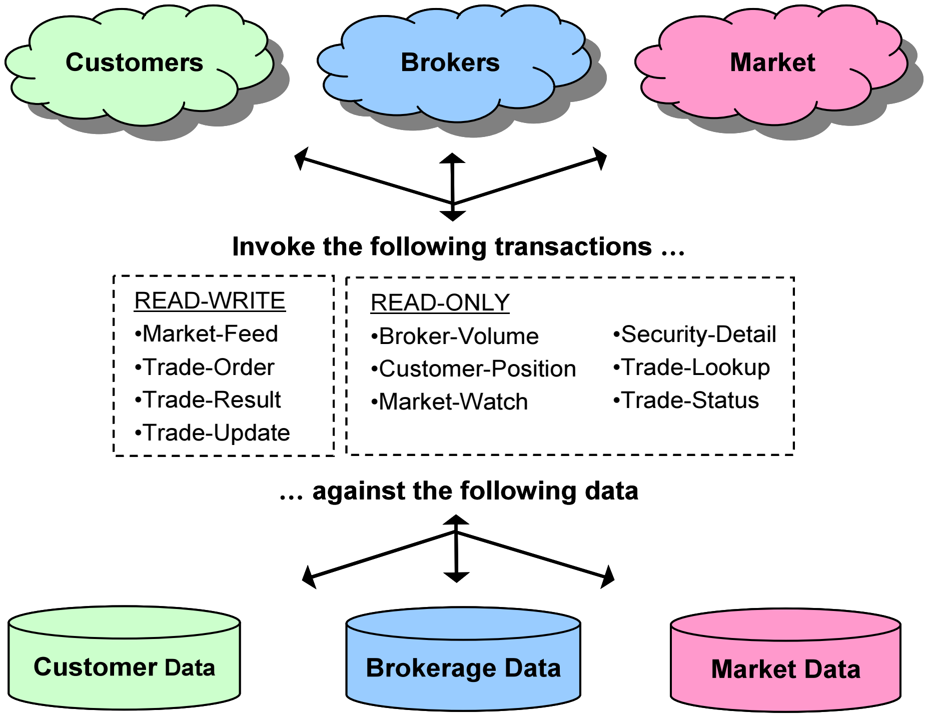
- **Satisfies the industry need for a benchmark that:**
  - Has a database-centric workload
  - Stresses virtualization layer
  - Moderate # of VMs, exercising enterprise applications
  - Healthy storage and networking I/O content; emphasizes I/O in a virtualized environment
  - *NOT* many app environments in an app consolidation scenario
- **Timely development cycle (1-2 years)**
  - Based on the *TPC-E* benchmark and borrows a lot from it
  - But is a different workload mix and the results cannot possibly be compared to *TPC-E* results
- **Results not comparable to other TPC benchmarks**
- **Generates information not covered by other benchmarks**

30000-foot view of TPC-V

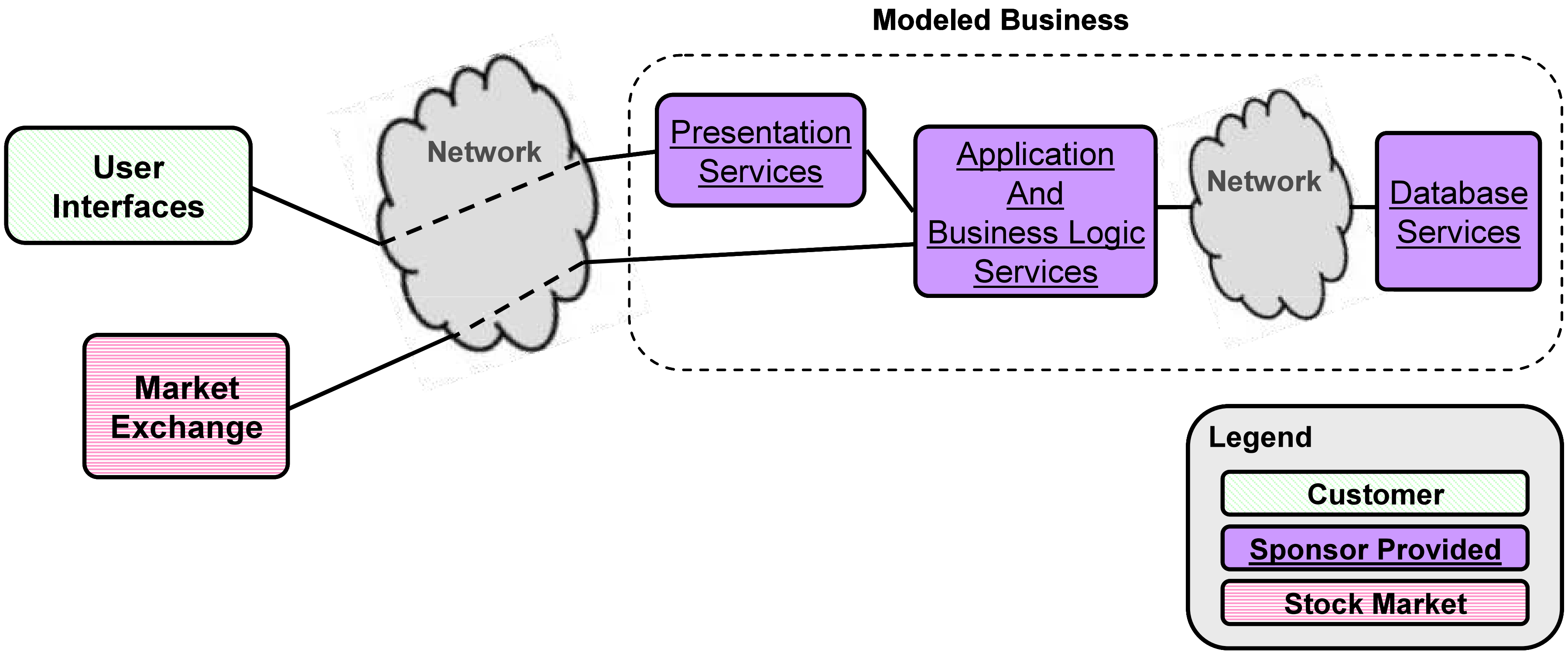
- **Rely on *TPC-E* as much as you can:**
  - EGen => Vgen
  - Schema and table definitions
  - Transactions
  - Application logic
- **Divide the *TPC-E* transactions into two groups**
  - VM 1 handles the OLTP transactions (and is CPU heavy)
  - VM 2 handles the DSS queries (and is I/O heavy)
  - The two VMs and their databases are not aware of each other
- **Variable # of VMs, tied to overall performance**
- **Varies the load dynamically during MI**
- **The last two properties make it unique and appealing since they emulate real-world**
  - But also challenging to implement

What is *TPC-E*

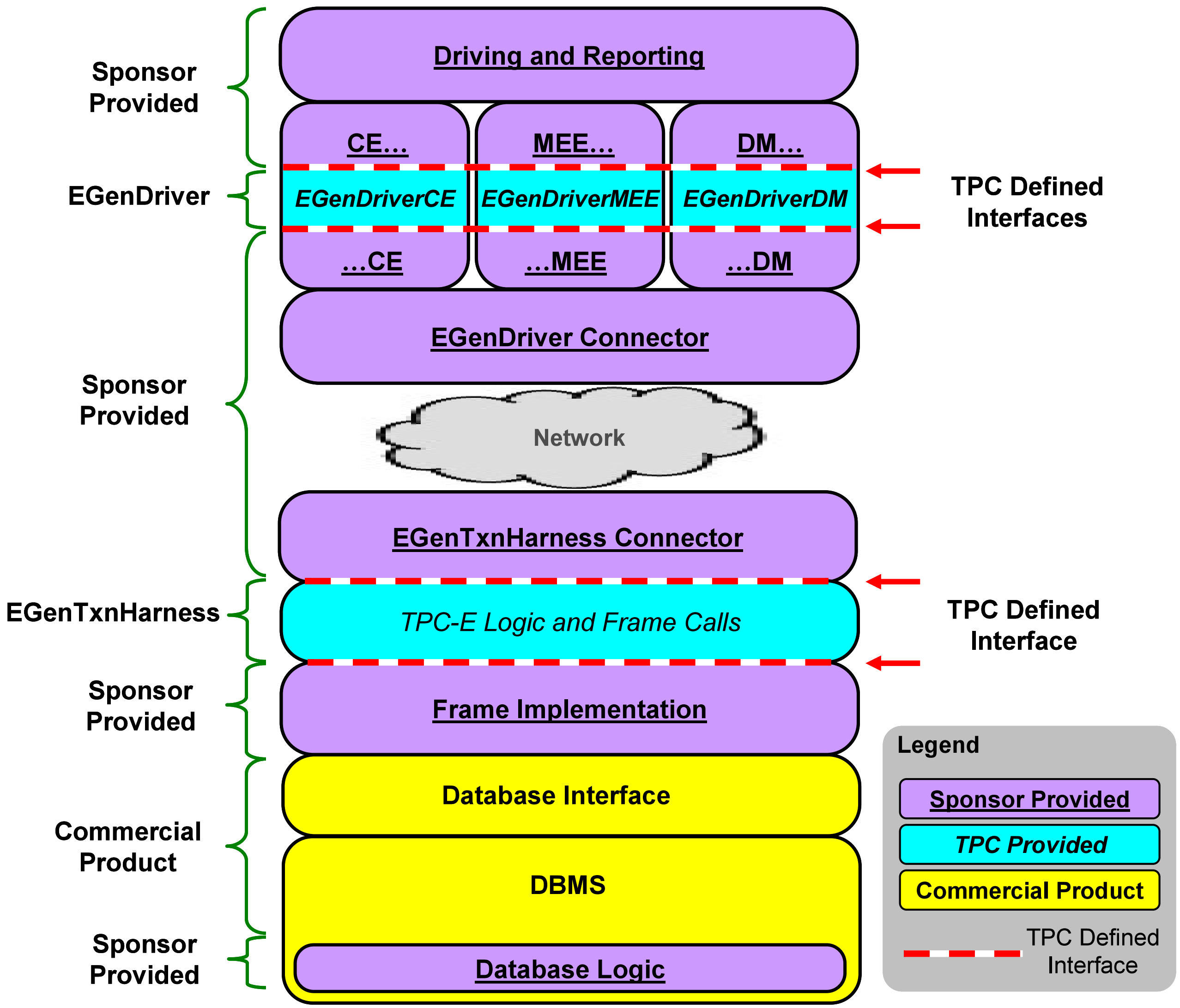
- ***TPC-E* is theTPC's latest OLTP benchmark**
  - More complex than TPC-C
  - Less I/O than TPC-C
  - A lot of the code is TPC-supplied
- **Models a brokerage firm**



Abstraction of the Functional Components in an OLTP Environment



Functional Components of *TPC-E* Test Configuration



TPC-V design considerations

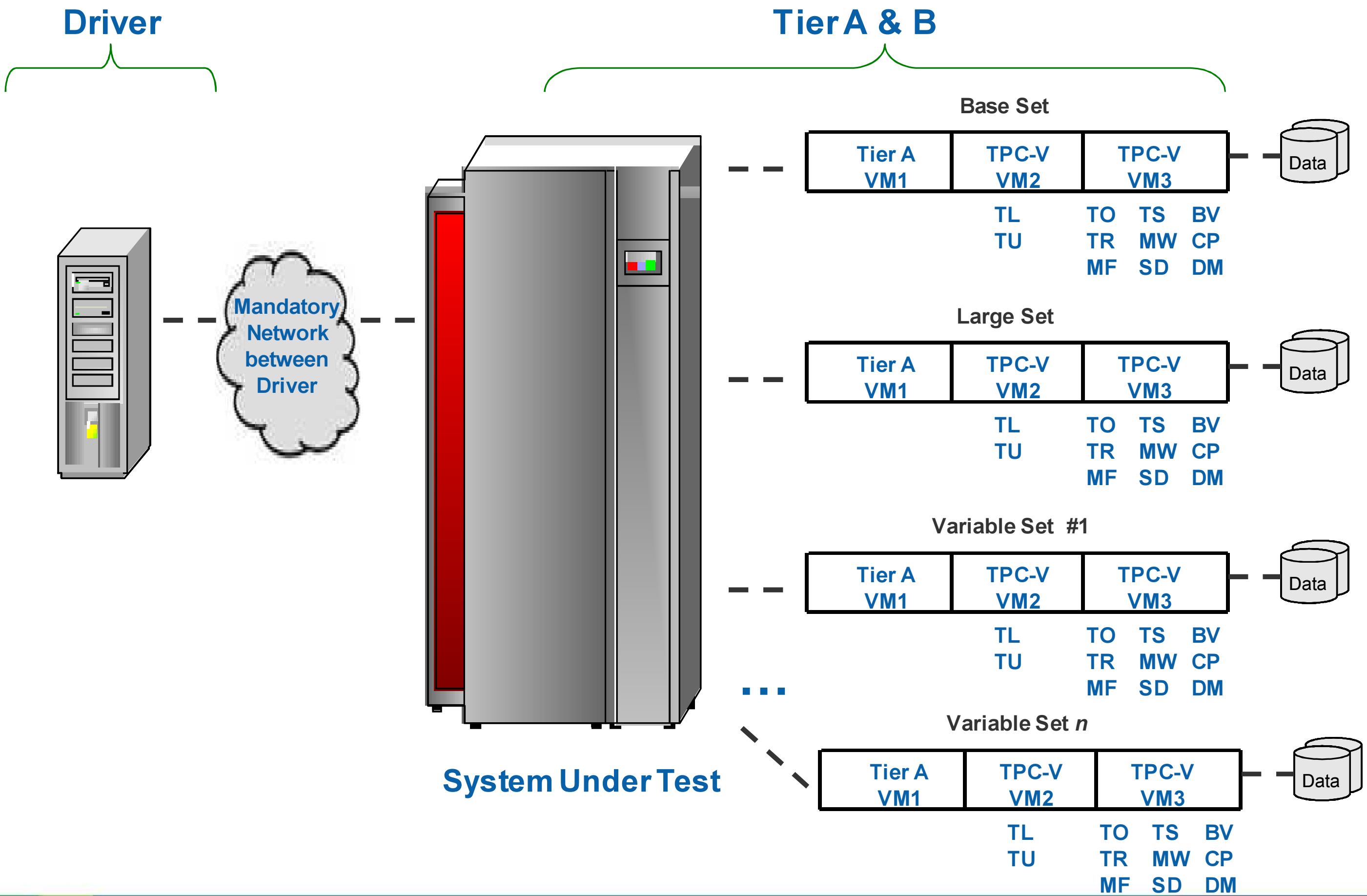
- **TPC-V does not specify the resource allocation (CPUs, memory, virtual disk and network …) of each VM**
  - Only the load (throughput) placed on each Set/VM
- **Devised a *Set* architecture:**
  - Different Sets on a system have different load levels
    - Because a virtualized host has to deal with the challenge of managing resources across VMs with varying demands
  - As throughput increases, TPC-V will have both more Sets and higher performance per Set
    - Because VMs on more power servers process more load
- **Chose not to define the load based on virtual processors or physical cores**
  - Not all cores are created equal

Set architecture in TPC-V

- **Set: a grouping of 3 or more Virtual Machines:**
  - One or more Tier A VMs
  - One or more transaction specific Tier B VMs that implement the I/O heavy transactions (TL and TU)
  - One or more transaction specific Tier B VMs that implement the CPU heavy trade transactions
- **At a given overall performance at the system level, the benchmark specifies:**
  - how many Sets of VMs
  - how much of the overall throughput is contributed by each Set

Sample Components of Physical Test Configuration

- A *TPC-E* SUT has 1 Tier A and 1 Tier B
- But each Set in a TPC-V SUT has 1 Tier A VM and 2 tier B VMs

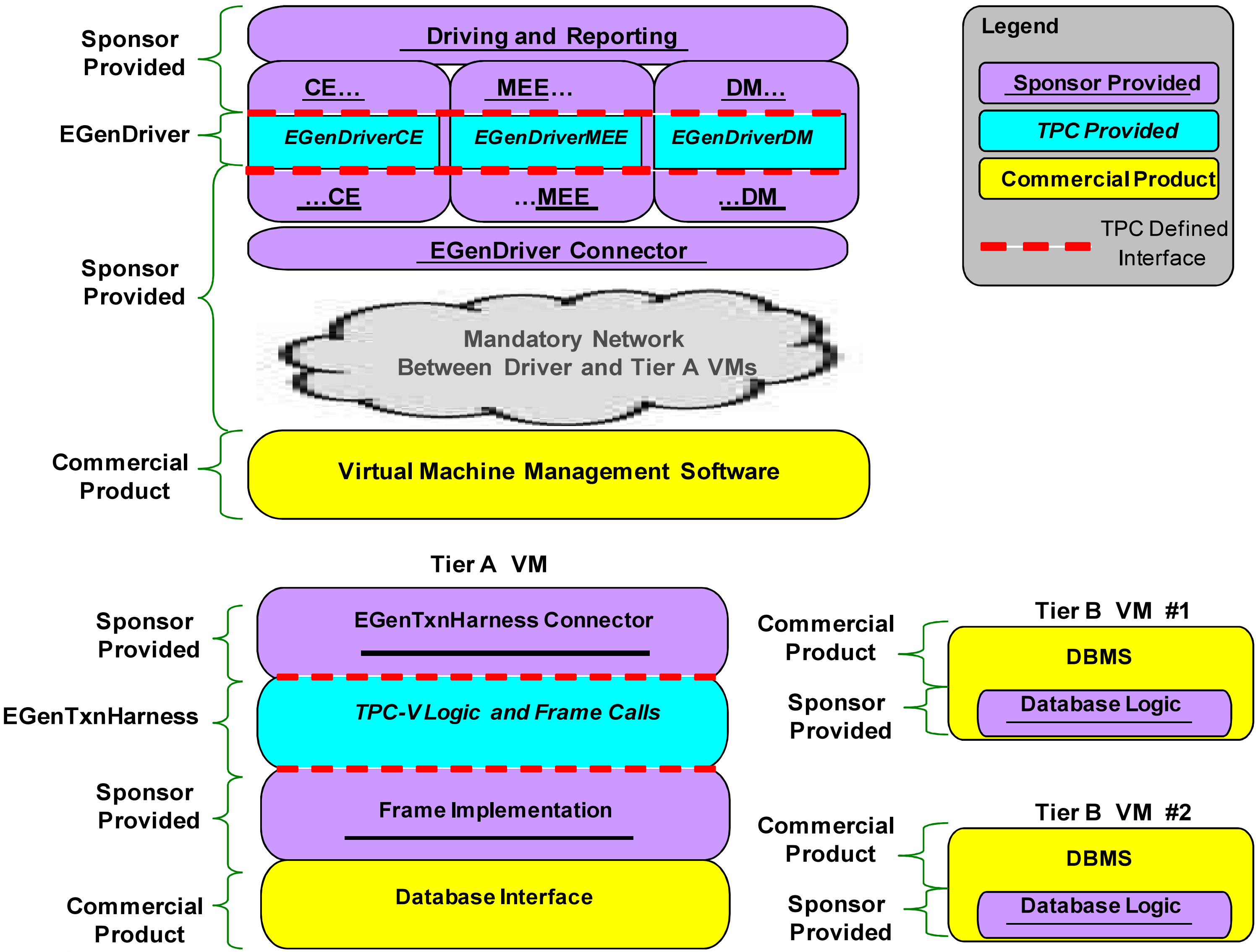


vmware

Set Architecture

- **More *and* larger VMs on more powerful servers**
  - A ***Base*** Set; a ***Large*** Set; N ***Variable*** Sets
  - The number of sets increases sub-linearly with the server's performance
  - Throughput of Sets increases sub-linearly with the server's performance
  - f(tpsV) = max(1, SQRT(45% * tpsV) / M + C)
    where M=SQRT(40) and C=-2
  - Exact details after prototyping
  - Some examples based on current formulas
    - 5 Sets, 133 to 900 tpsV on a 2000-tpsV SUT
    - 19 Sets, 84 to 14,400 tpsV on an 32000-tpsV SUT

Functional Components of TPC-V Test Configuration



TPC-V SET/VM/VP Sizing Worksheet

SUT Target tpsV=>	1,000	2,000	4,000	8,000	16,000	32,000
min Variable Sets	-	2.0	4.0	6.0	10.0	16.0
max Variable Sets	1.0	3.0	5.0	7.0	11.0	17.0
Base Set tpsV	150	300	600	1,200	2,400	4,800
Large Set tpsV	450	900	1,800	3,600	7,200	14,400
Variable Set tpsV	400	800	1,600	3,200	6,400	12,800
Total Sets (S+L+M)	3	5	7	9	13	19
Base Set VMs	3	3	3	3	3	3
Large Set VMs	3	3	3	3	3	3
Medium Set VMs	3	9	15	21	33	51
Max Total VMs	9	15	21	27	39	57
Variable Set 1 tpsV	400	133	107	114	97	84
Variable Set 2 tpsV	-	267	213	229	194	167
Variable Set 3 tpsV	-	400	320	343	291	251
Variable Set 4 tpsV	-	-	427	457	388	335
Variable Set 5 tpsV	-	-	533	571	485	418
Variable Set 6 tpsV	-	-	-	686	582	502
Variable Set 7 tpsV	-	-	-	800	679	586
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Variable Set 17 tpsV	-	-	-	-	-	1,422
Total Variable Set tpsV	400	800	1,600	3,200	6,400	12,800

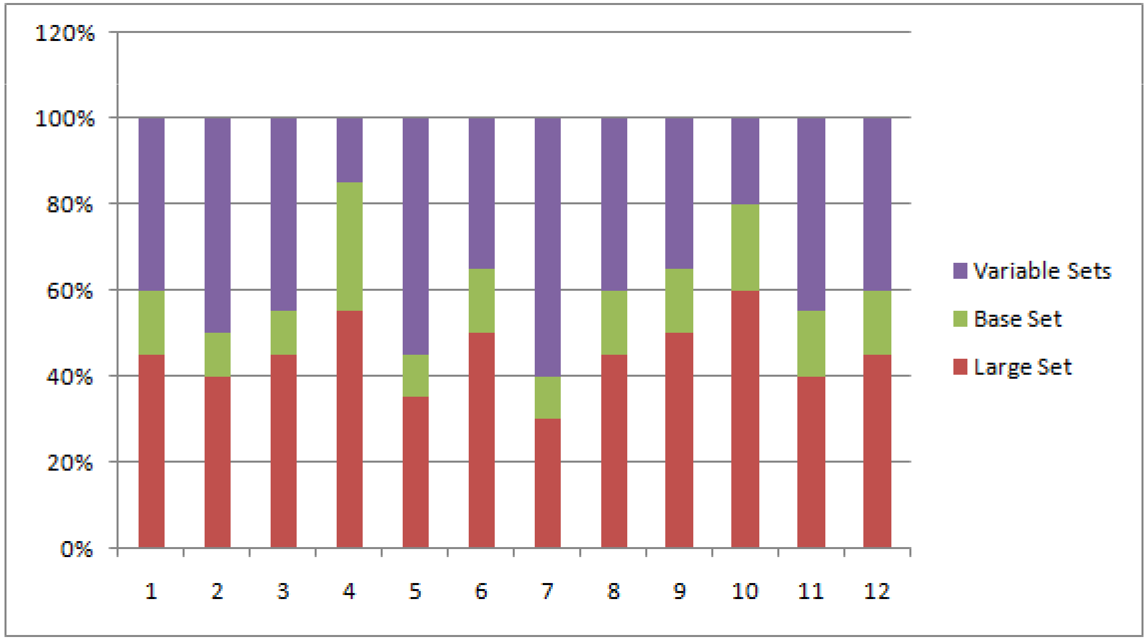
Elasticity

- **Performance benchmarks are typically measured in “steady state” with a constant throughput**
  - OK for a single application, but not for a virtualized server.
- **The load placed on a Set will vary during the benchmark run, simulating elasticity**
  - exercise the resource allocation properties of the VMMS based on workload demands
  - One of the major properties that draws customers to virtualized servers
- **It’s all about the cloud!**

Elasticity

- **Load of VMs varies by 2X to 4X during Measurement Interval**
  - Elasticity
  - Oversubscription

Period	Large Set	Base Set	Variable S
1	45%	15%	40%
2	40%	10%	50%
3	45%	10%	45%
4	55%	30%	15%
5	35%	10%	55%
6	50%	15%	35%
7	30%	10%	60%
8	45%	15%	40%
9	50%	15%	35%
10	60%	20%	20%
11	40%	15%	45%
12	45%	15%	40%
Average	45.0%	15.0%	40.0%



Benchmark Development Status

- **Working Group has evolved into a Development Subcommittee**
  - 11-13 member companies
  - Actively meeting on weekly conf calls
- **Submitted a draft spec in June**
- **Worked through a lot of thorny issues**
- **It looks like the benchmark will be a winner if code developers can come up with a driver (RTE) than can:**
  - Drive multiple Sets with a deck of cards method
  - Vary the load to each Set dynamically
- **Have started prototyping**
  - A lot of numerical values might change

Why not simply a virtual *TPC-E*?

- **The council asked us specifically for a non-comparable-results benchmark**
- **Regardless, we'd need a whole new benchmark to incorporate:**
  - Multiple VMs in a single SUT, single FDR
  - The number of VMs has to be decided by the spec to make all results comparable
  - A business model based on a variety of applications rather than a single application
  - Measure important attributes of virtualization
    - Multiple VMs
    - VMs of different sizes
    - Elasticity
    - Oversubscription

Why not a generic framework for all TPC benchmarks?

- A generic framework development process will be much longer than the 1-2 years the council has asked for
- TPC-V can be the starting point for a generic framework that can be applied to other TPC benchmarks

What about blades/scale out/clusters?

- **The spec needs to allow growth in technology**
- **Avoid an “embarrassingly parallelizable” workload**
  - Elasticity + large Set size
  - Properties inherited from *TPC-E* schema
- **Blades or clusters not prohibited or discouraged**

Challenges

- Schedule
- Dependence on member companies committing development resources
- Throwing problems over the wall at the driver/RTE
- Usual committee issues

Acknowledgements

- The TPC-V benchmark is being developed through the efforts of contributors from more than a dozen companies. Much of the material reported here is the result of the work of the committee members as a whole, rather than the authors.
- We would in particular like to acknowledge the contributions of John Fowler, Karl Huppler, Doug Johnson, Matthew Lanken, Jody McCoy, Arno Mihm, Pete Peterson, Francois Raab, Cecil Reames, Jamie Reding, and Wayne Smith.