2009 Next-Generation Data Centers

August 13, 2009 – San Francisco

SYBASE®

Virtualizing Resources in Production Databases

It takes more than hardware and virtual machines to exceed SLAs!

Peter Thawley

Senior Director / Architect

CTO Group, Sybase

Agenda

- Next Generation Data Centers
  - Key Characteristics from a Database Perspective
- A Quick Tour of Virtualization
- Virtualization Meets Production – Assumptions
- Virtualizing Database Processing – It's All About SLAs!
  - Memory Management
  - CPUs, Cores, and Strands in a Symmetric Multiprocessor Architecture
  - Server Nodes / DB Instances in a Shared Disk Cluster Architecture
- Summary

Next-Generation Data Centers

Key Characteristics from a Database Perspective

- **Elasticity** – dynamically grow & shrink resources through DBA tools or programmatic interfaces using policy-driven load balancing with no application code required
- **Low TCO** – minimize CAPEX expenditures from higher utilization and virtual resource management
- **Multi-tenancy** – simplify operations while ensuring SLAs at a granular level through elasticity and isolation
- **Reliability** – continuous availability with policy-driven fail-over and fail-back with no application code required
- **Security** – Secure infrastructure using traditional enterprise database techniques and granular (column-level) encryption with no application code required

Virtualization – What is it?

"In computing, virtualization is a broad term that refers to the abstraction of computer resources."

About.com

"Virtualization is a technique for hiding the physical characteristics of computing resources to simplify the way in which other systems, applications, or end users interact with those resources."

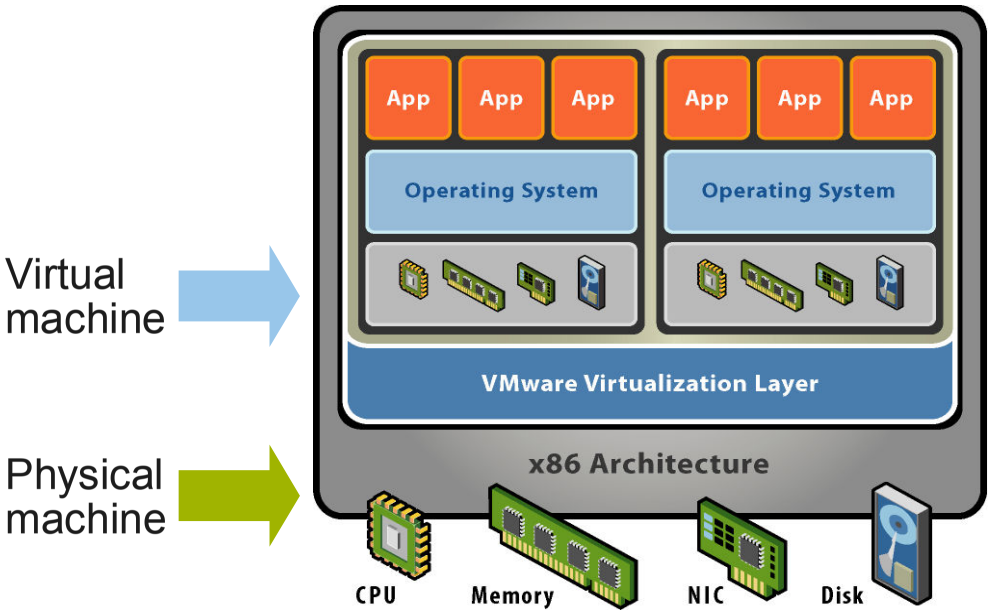
SYBASE®

Virtualization Comes in Many Forms

- Mature Concept in many Different Areas of Computing:
  - ***Virtual Memory*** allows the OS to "fool" the application into thinking it has more memory than it really does;
  - ***Platform virtualization*** abstracts an operating system from the underlying HW platform resources;
  - ***Resource virtualization*** abstracts specific system resources:
    - ***Storage virtualization*** abstracts physical location to allow flexible movement of files for load balancing, tiering, migration, etc.
    - ***Network virtualization*** abstracts network addresses and domains to allow applications
  - ***Data virtualization*** abstracts differences in data stores (RDBMS, files, web services, etc) and location to provide applications with consistent syntax and semantics
  - ***Application virtualization*** allows the hosting of individual applications on alien hardware/software

Virtualization – the Defacto Usage:*

- A level of indirection between hardware and software
- Virtual Machines: a full abstraction of a physical machine
  - Safely and efficiently multiplex virtual hardware on physical hardware

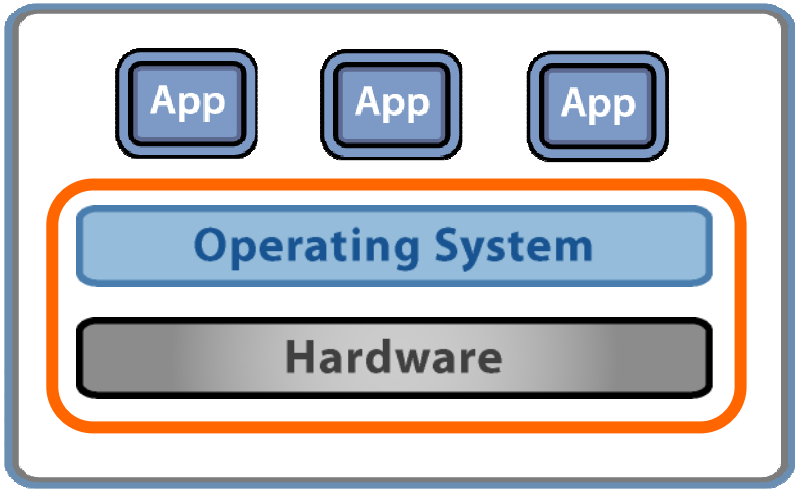


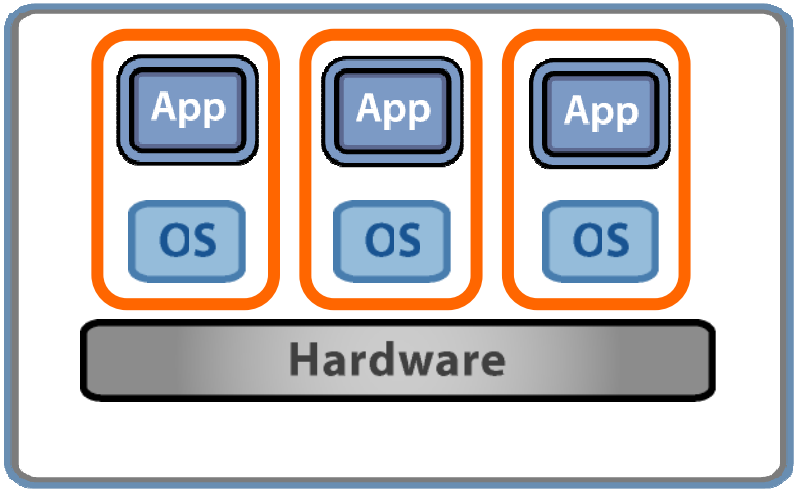
- Benefits: resource sharing, portability, mobility, compatibility

* Slide content courtesy of VMware®, an EMC² company.

SYBASE®

Server & OS Virtualization*





- **Before Server Virtualization:**
  - Single OS image per machine
  - Software & hardware tightly coupled
  - Running multiple applications on same machine often creates conflict
  - Underutilized resources
  - Fault/security vulnerability

- **After Server Virtualization:**
  - VMs **break 1-to-1 dependency** between OS and HW
  - Manage OS and application as single unit by **encapsulating** them into VMs
  - **Hardware-independent:** they can be provisioned anywhere
  - Strong **isolation** between VMs

* Slide content courtesy of VMware®, an EMC² company.

SYBASE®

Virtualization Meets Production

General Assumptions about which Applications / Systems are Candidates to Virtualize

- Applications / Services **Consume "Small" % of System Capacity**
  - Adequate Capacity of System to "Timeshare" Simultaneously
- Applications / Services **Execute for "Short" Durations**
  - Grid-style Application Partitioning – May Consume Entire System Capacity, but only for Limited and Mutually Exclusive Amounts of Time
- Applications / Services **Well-Behaved** in a Virtual Machine
  - Shares Resources "Well with Others"…
    - No "Idle Spinning" or "Wasting" of Resources Others Could Use
- Applications / Services **Not Sensitive to [Variable] Latency**
  - Performance Overhead Diminishing, but still Present
  - Virtualization at Odds with Deterministic Performance

Production Database Instances

Are Production Databases Candidates for Virtualization?

- As Always, It *Depends* …
  - Smaller, Non Business-Critical Production Database Systems are Being Successfully Virtualized Today
    - Sybase running 100's of production database instances today
- But Production Databases for Business-Critical, Tier 0/1 Applications *Generally* Don't Fit Previous Assumptions …
  - Utilization Often too High (as a % of Capacity)
  - Multi-User and Performance Requirements Often too Significant
  - Predictability Often Valued More than Cost of Resource Sharing
  - Risk of "Yet Another Moving Part"
- *Question:* Aren't these Business-Critical, Tier 0/1 Applications Exactly the Ones that Need the Benefits of Virtualization Most?
  - Flexible Resource Management that Allows Resources to be Moved Around the System in Response to Capacity Needs …

Virtualizing Resources within the Database

Is this Really Useful?

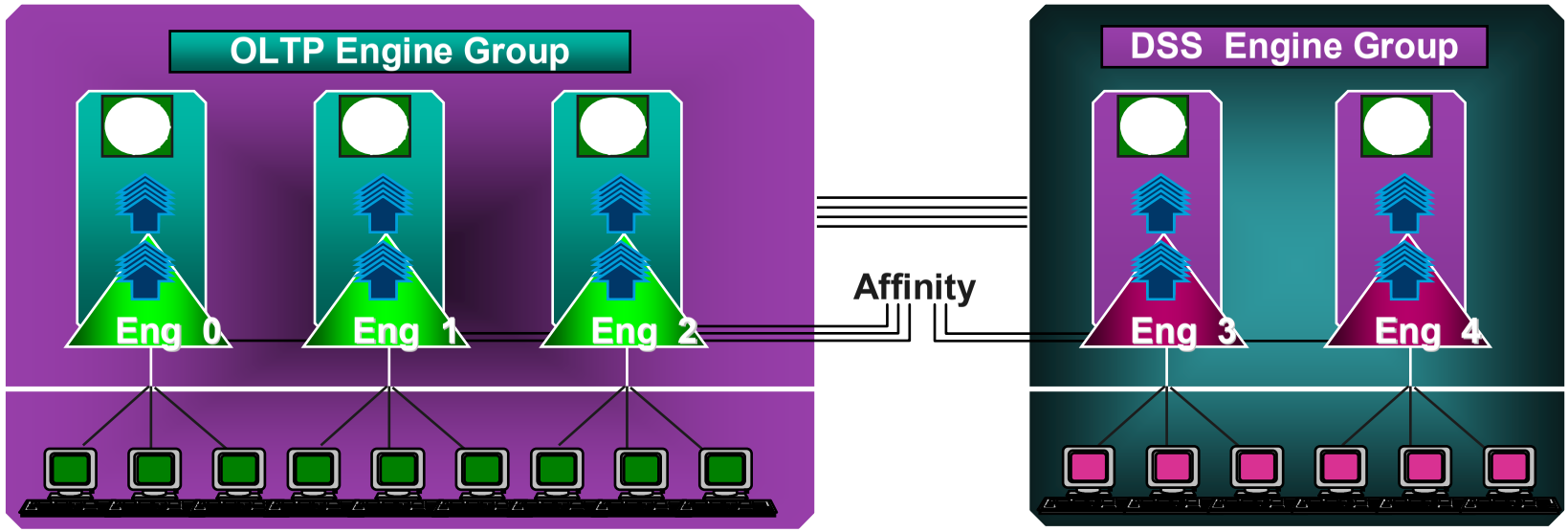
- These Databases Often Serve Multiple Applications or Workload Profiles
  - Often Simultaneously
  - Often with Different SLAs
- Consider a Trading System with Concurrent OLTP and DSS
  - Risk of Starving OLTP Application High
    - OLTP short duration execution profile before blocking
    - DSS, especially in large memory configurations, don't block often
- How can you "carve up" resources and dynamically bind resources to application?

Resource Management in the 21st Century

- Multiple applications per server is normal
  - Need to ensure “service levels” to end-users
  - Resource requirements vary by application, module, and even time of day!
- Batch & maintenance windows shrinking
- Need to provide dynamic flexibility in customizing resource allocation to meet diverse and changing needs
  - “Transaction Processing” versus “Batch Processing”
  - Easy to saturate large hardware with Operational Decision-support so how do you minimize operational risk
    - Software parallelism is a brute-force approach to good response time

Virtualizing CPUs, Cores, and HW Strands

Virtual Process Management



- Customize resources tailored to application environment
  - Affinity threads, procedures, or application to engine(s)
  - Task priorities per thread, application, or stored procedure
- Predictable performance for mixed workloads
  - Guaranteed Resource Reservation to Ensure Capacity for SLA's
  - Low-priority work doesn't impact high priority work
- Dynamic "Application to Resource" mapping
  - Move Resources Between Applications As Needed

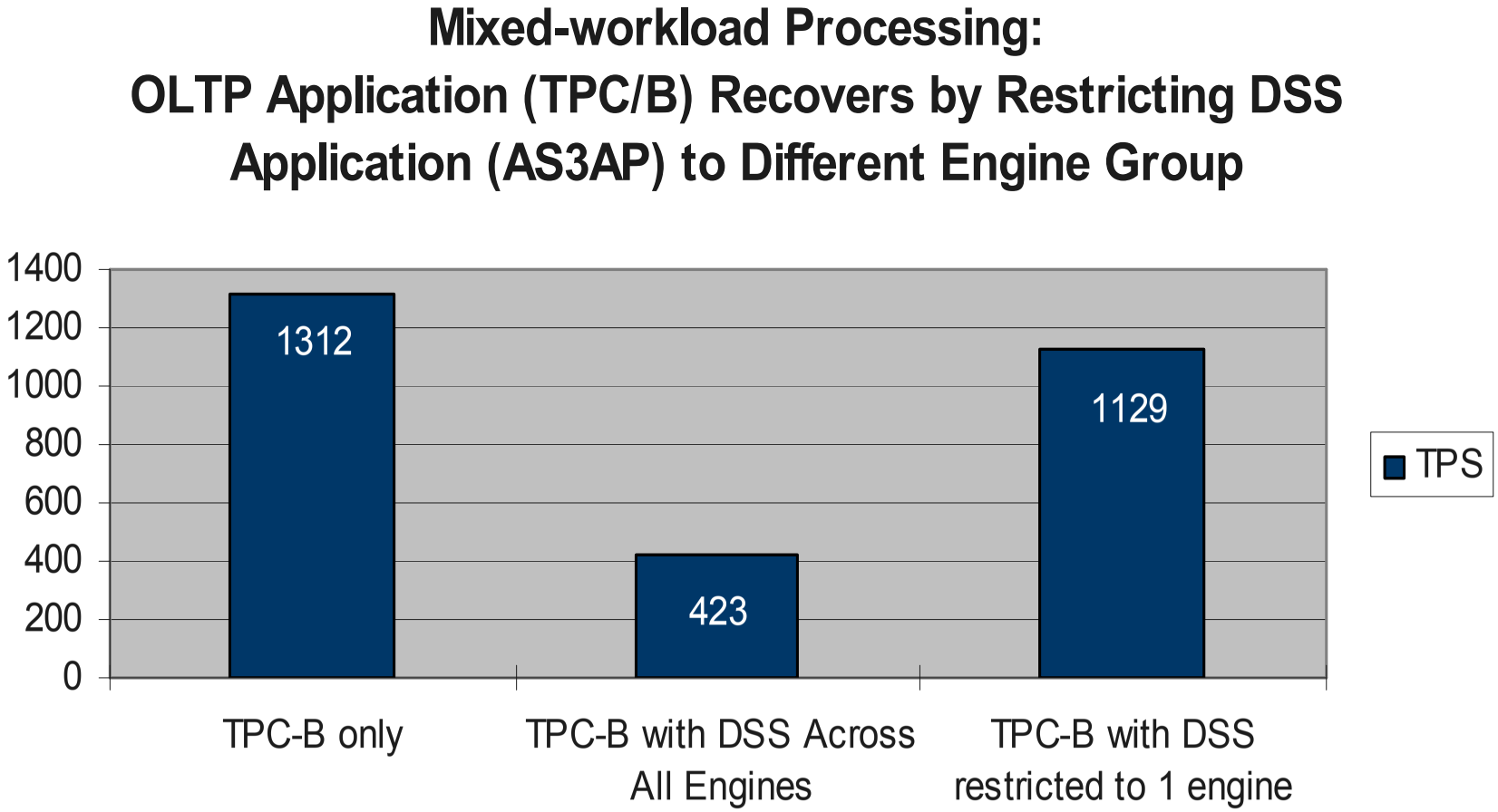
SYBASE®

Virtual Process Management

Key Concepts

- "Engine" Groups
  - Method of tying together specific engines of an Adaptive Server into a named group
  - Provides finite amount of CPU for consumption
- Execution Priorities
  - Provides for execution of some entity before others
- Execution Classes
  - Method by which Engine Groups and Priorities are assigned to applications, threads, and stored procedures

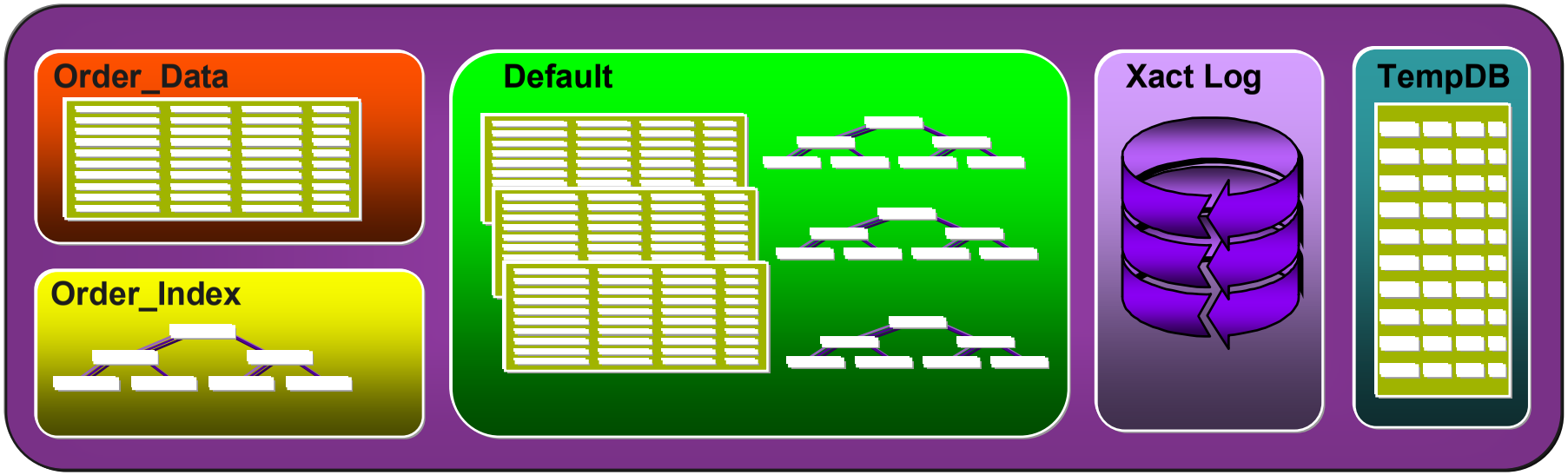
Virtual Process Management – In Action



SYBASE®

Virtualizing Database Cache Memory

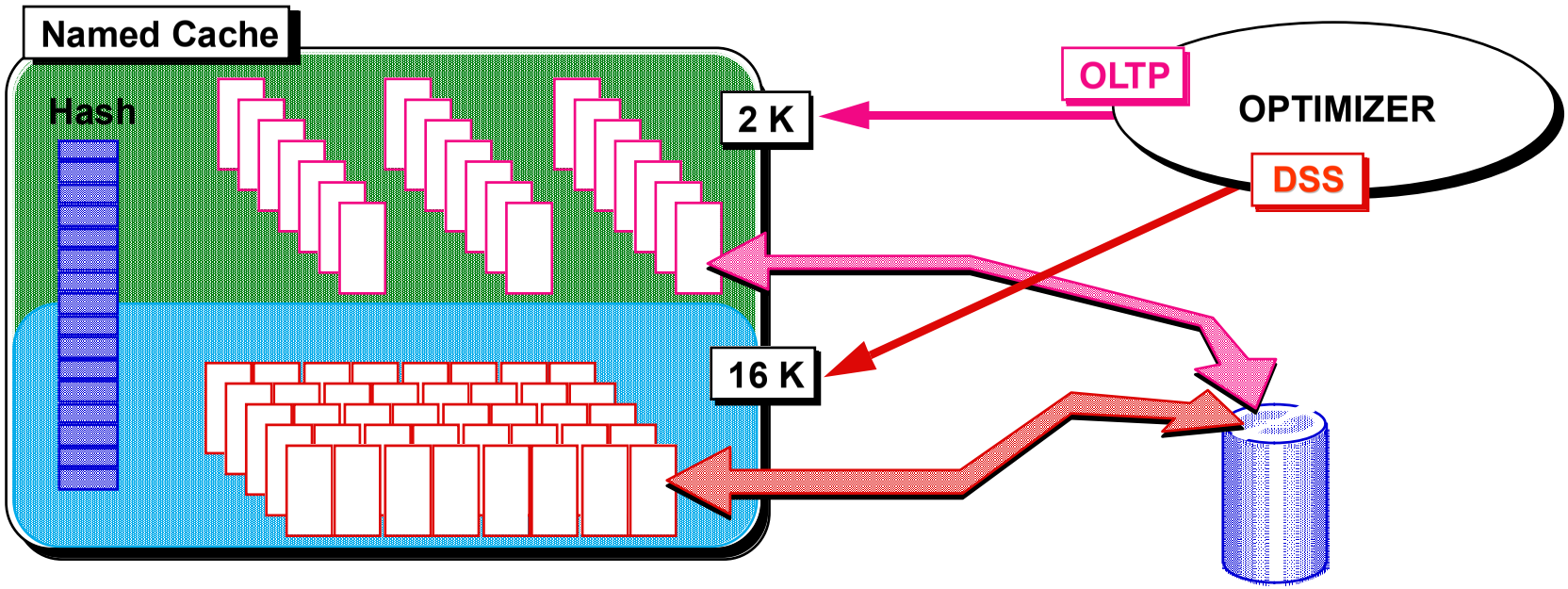
Treat Memory as an Optimizable Resource



- Database, Table, & Index Assignments to Specific Caches
- Dynamic Object to Cache Assignments & Memory Movement
- Reduction in SMP Contention on Synchronization
- Complete Control over Resource Consumption by Object
  - Only Rational Way to Support Large Memory
  - Ensure SLA's for Key Transactions with Guaranteed Resource Reservation

SYBASE®

Variable Block-size Disk I/O

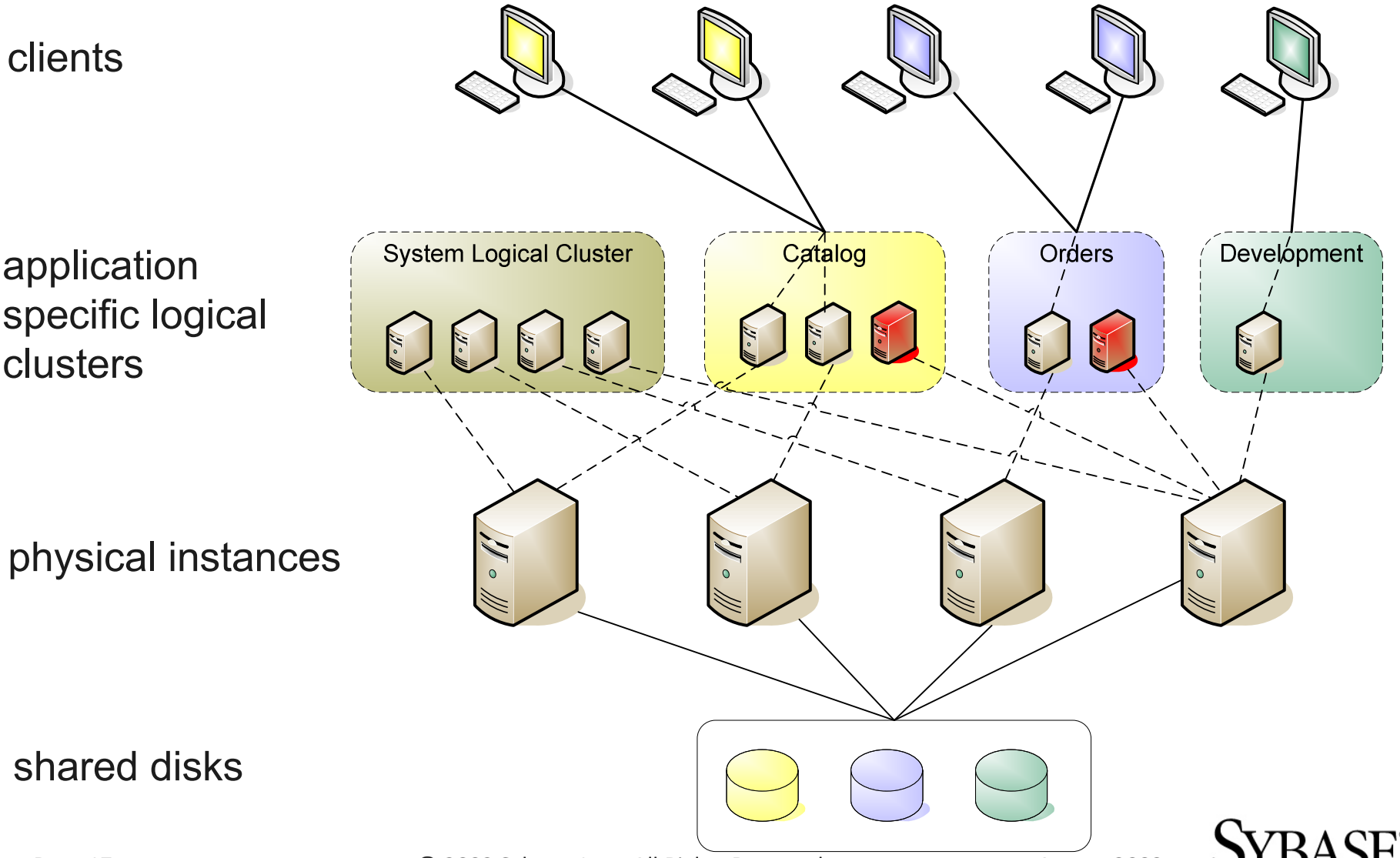


- Configure any Cache with Multiple Pools of I/O Buffers
- Supports I/O Sizes From 2Kb up to 128Kb
  - Page I/O vs Page Size vs. Extent I/O (8 pages)
  - Large Buffer called a MASS
- Optimizer-Driven to Balance System Needs & Resources
- Huge Increase in DSS, Load, and Text/Image Performance

SYBASE®

Virtualizing Nodes in a Cluster

Logical Clusters



Cluster Workload Manager

- Workload Manager - subsystem built into ASE
- Enables cluster management on an application basis
  - Routing of connections within a cluster based on application
  - Differing load balancing rules and load reporting based on application
  - Availability models (1+1, N+1, N+M, etc.) set at app level
  - Administrative failover, failback, offline, and online at the application level
    - Planned downtime with applications in mind.
  - True resource reservation – instances can be exclusively reserved for specific applications

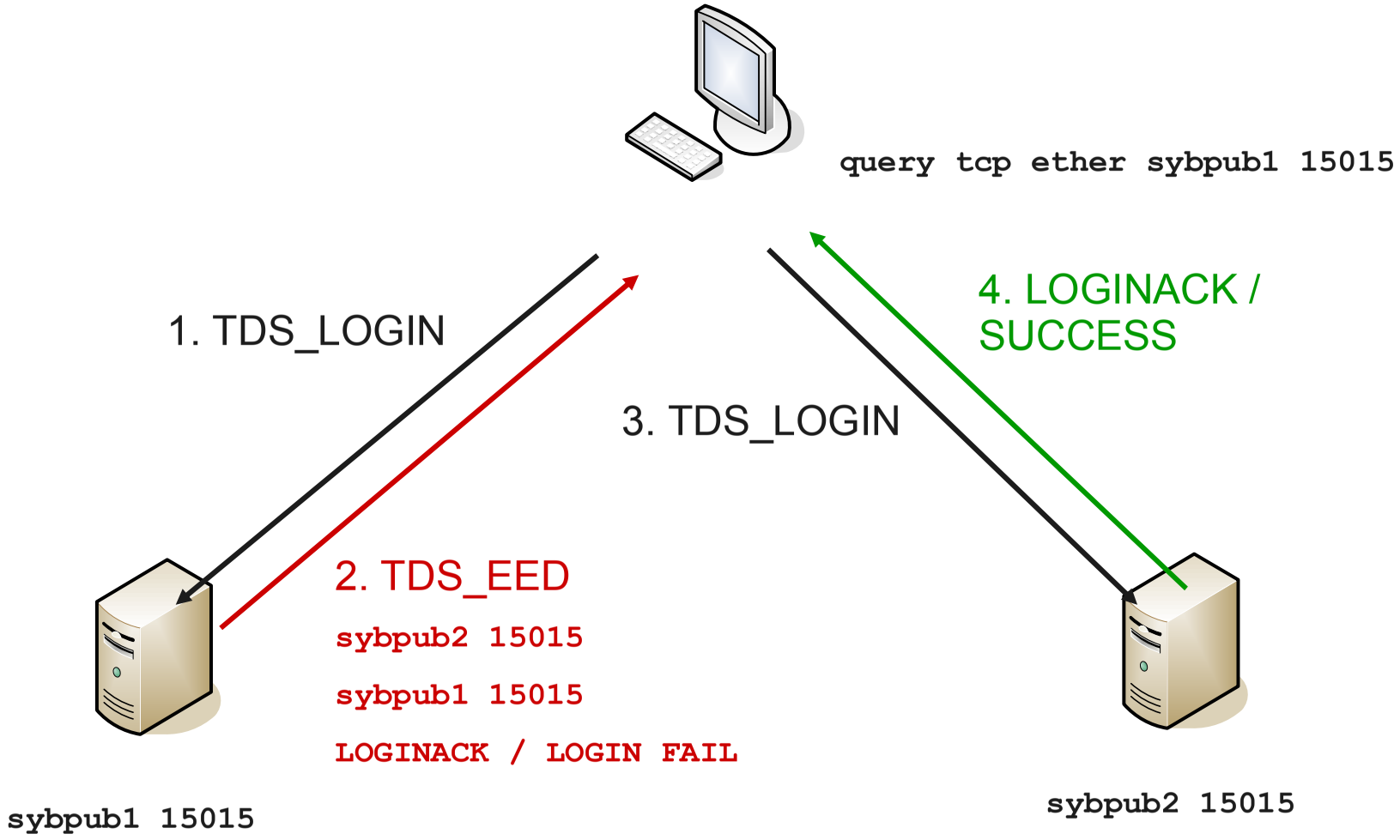
Logical Clusters

- Logical clusters are the primary entities in the workload manager subsystem
  - Used to represent an application or application partition
  - M:N relationship with instances
    - Logical cluster can run on many instances, many logical clusters can run on a single instance.
  - “base” instances are where they run by default
  - “failover” instances where they should run in the event of a failure
    - These can be grouped and prioritized
  - Numerous configurable attributes control behavior
  - Managed via online, offline, failover, failback commands

New Client-side API Technologies

- Workload Manager utilizes three new technologies in the client drivers
  - Connection (login) redirection
    - During login, the server instructs that client driver that it should connect elsewhere
  - Extended HA Failover
    - During initial connection server publishes a list of failover addresses to the client. ASE can update this list at a later time.
    - Single *hafailover* entry in directory service is replaced / augmented by the server provided list.
  - Connection Migration
    - Existing client connection is migrated to another server

Login Redirection



SYBASE®

Connection Migration

- What it is:
  - Ability of ASE to gracefully migrate an existing client connection and its context to another instance in the cluster.
    - Contrast with connection redirection which is used on incoming connections that have no context
- Why it is useful:
  - Dynamically distribute load across the cluster by moving existing connections to less busy instances
  - Gracefully failover, failback, migrated, or stop a logical cluster on a given set of instances
- How it works:
  - ASE identifies that a migrate-eligible connection should be moved.
  - Connection's context is packaged and sent to other instances.
  - ASE and the client take part in a migration dialog whereby ASE sends the client an ordered migration list.
  - The client connects to the migration target, its previous context is retrieved and unpackaged

Summary

- Production Databases Require More than HW and VM's to Gain Value from Virtualization.
- Key Concept is Internal Virtualization of both SMP & Cluster Hardware Resources and Dynamic Assignment to Applications
  - Instances: Nodes of a Shared Disk Cluster
  - Engines: CPUs / Cores / HW Strands
  - Memory: Caches / Pools
  - Network: Listeners
  - Disks: Devices
- Why?
  - Predictability and Manageability of Service Levels
  - Flexibility to Move Resources as Demand Dictates

Resources to learn more

- For more information, visit
  http://www.sybase.com/clusters

- IDC Research paper on topic
  http://www.sybase.com/IDCDatabaseVirtualization

- Send me an email with your thoughts and questions

Peter Thawley Senior Director / Architect, CTO Group
peter[dot]thawley[at]sybase[dot]com

SYBASE®

Thank you!

Questions?