A Technical Look Inside ASE 15.7's "Hybrid-Threaded" Kernel

K^{21} – ASE's Kernel Design for the 21^{ST} Century

DIAL IN NUMBER:
866.803.2143
210.795.1098
PASSCODE: SYBASE

A Technical Look Inside ASE 15.7's "Hybrid-Threaded" Kernel

Your host…

Terry Orsborn

Product Marketing Manager

Our speaker today…

Peter Thawley

Senior Director/Architect, CTO Office

HOUSEKEEPING

Questions?

Submit via the ‘Questions’ tool on your Live Meeting console, or call 866.803.2143- United States or 210.795.1098 - International with the Password: SYBASE during the Q&A segment

Presentation copies?

Select the printer icon on the Live Meeting console

A Technical Look Inside ASE 15.7's "Hybrid-Threaded" Kernel

K^{21} – ASE's Kernel Design for the 21^{st} Century

Peter Thawley

Senior Director / Architect, CTO Office

November 2011

AGENDA AND ACKNOWLEDGEMENTS

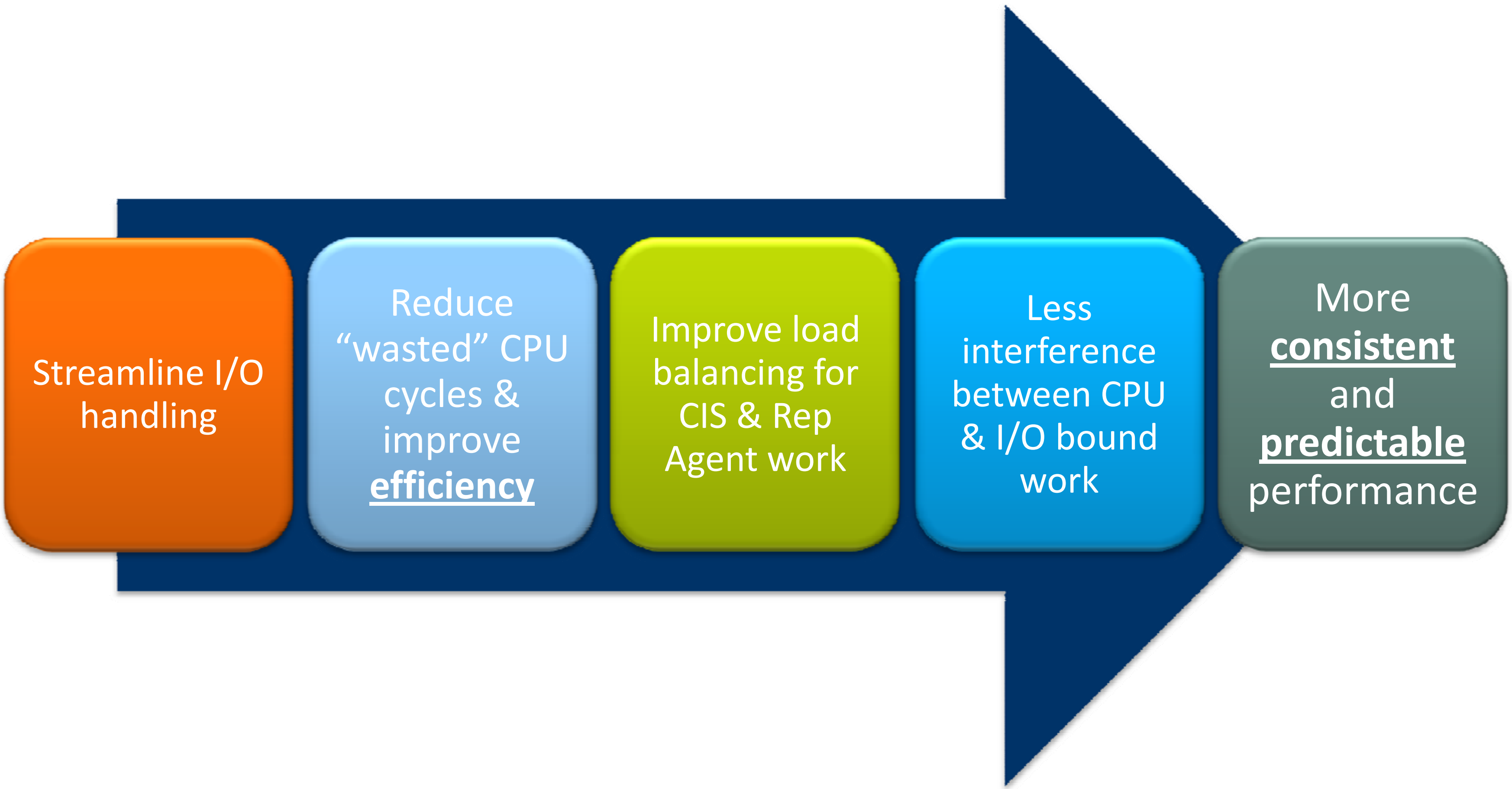
- Architectual Introduction
- Workloads
- Configuration and Tuning
- Interpreting sp_sysmon

Content Originally Developed for TW 2011 by:

David Wein, Technical Director, ASE Engineering

MOTIVATIONS

What the threaded kernel brings to the table.



FOUNDATIONS OF ASE ARCHITECTURE

Core principles remain unchanged. If you know the old kernel, you know the new kernel.

Each connection is given a SPID / KPID…we call this a **task**

→

Tasks are **internal** to ASE. The operating system does not see them.

→

Tasks run on **engines**, which are analogous to CPUs

→

A **scheduler** within ASE selects the task to be run by the engine

→

Engines communicate via **shared memory**

ONE ADAPTIVE SERVER, TWO KERNELS

Process Kernel

- Pre-15.7 kernel (except Windows)
- Each engine is a separate process
- Retained in 15.7 for risk mitigation

Threaded Kernel

- Default kernel for 15.7
- Each engine is a thread of a single process
- Additional threads for handling I/O, etc.
- ASE on Windows has always been thread based

THE PROCESS KERNEL

ASE on Unix and Linux as you've always known it.

- Each engine is a separate operating system **process**
- Disk I/O issued by one engine must be completed by that engine
- All network I/O for a connection must be performed on the engine that "manages" the connection
- Rep Agents and tasks using CIS cannot migrate between engines.

THE THREADED KERNEL

The default kernel for ASE 15.7

Engines are **threads** of a **single process**

No Engine to I/O ownership

Engines only need to run user tasks…no need to handle I/O

Additional non-engine threads handle I/O polling and completion

Rep Agent and CIS tasks no longer bound to engines

A NOTE ABOUT COMPATIBILITY…

The threaded kernel can be adopted without any changes to applications, and with minimal changes to configuration settings.

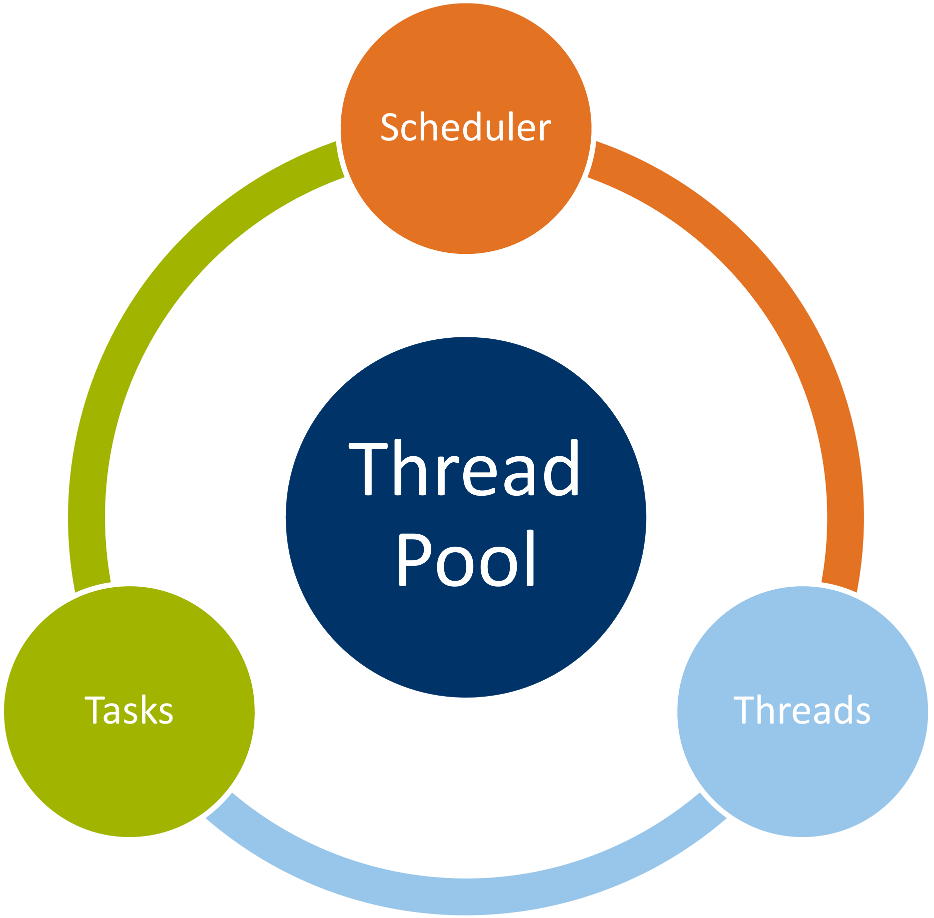
The new kernel was delivered to the SAP Business Suite development team midstream. No changes in the application layer were made to use the new K^{21} kernel!

SYBASE | An SAP Company

TASKS & THREAD POOLS

The new kernel is organized around thread pools

- All threads live in a thread pool
- All work is done by tasks, which are assigned to a thread pool
- Threads can only see tasks in the same pool
- Tasks will only be scheduled by threads in the same pool



THREAD POOLS

Three system provided pools are always present.
User created thread pools are possible, discussed in the configuration section

syb_default_pool

- Engines live here
- All kpid / spid related tasks live here
- Size this pool to get multiple engines

syb_system_pool

- I/O threads
- Signal thread
- Misc low CPU threads
- Size not directly user configurable

syb_blocking_pool

- Threads make long-latency calls on behalf of database tasks
- Calls can block in these threads instead of blocking engines

I/O POLLING

I/O handling is the biggest difference between kernel modes

Process Polling

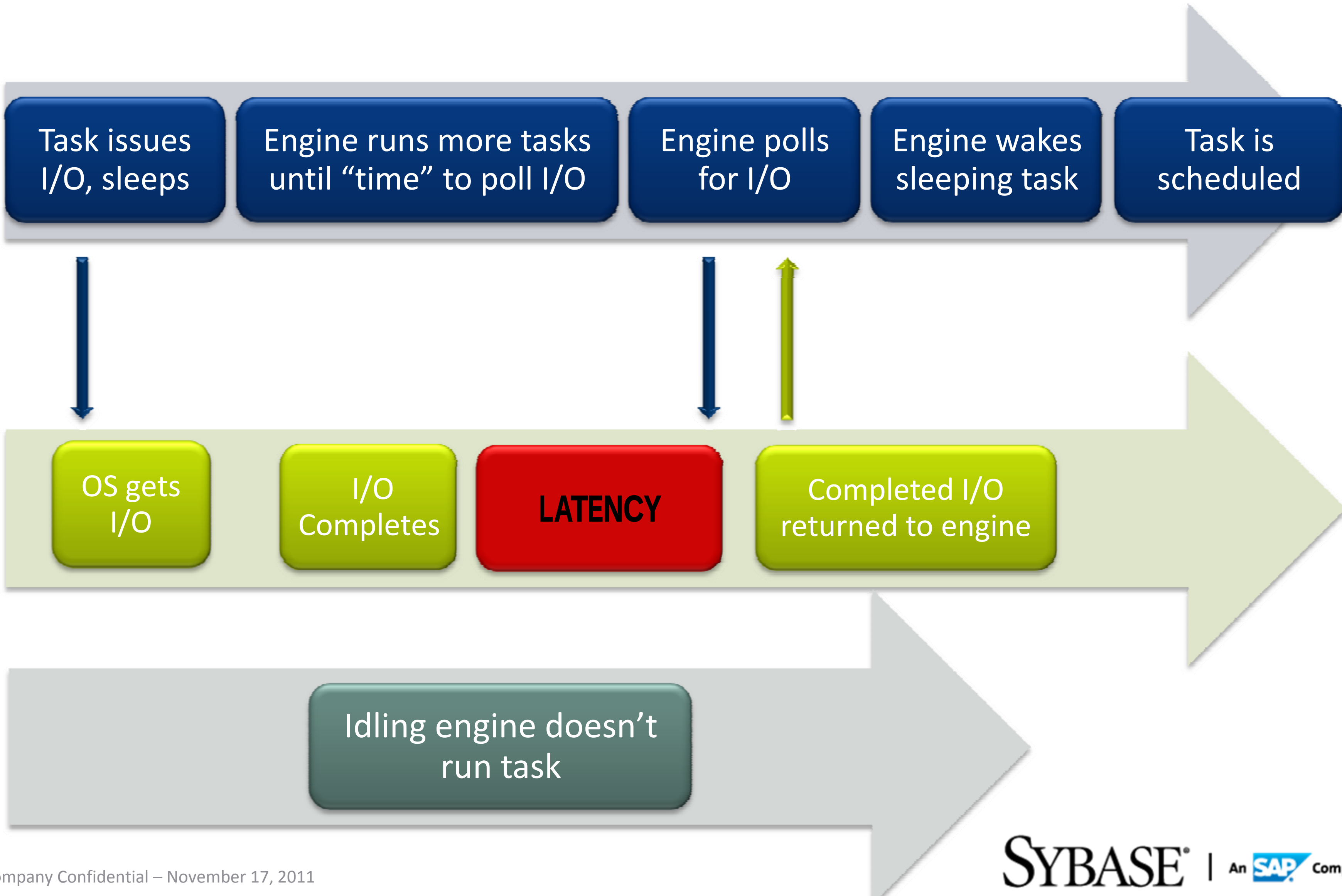
- Engines poll for I/O "inline" between running tasks
- The engine that issued the I/O must complete it.
- Idling engines cannot help a saturated engine complete its I/Os

Threaded Polling

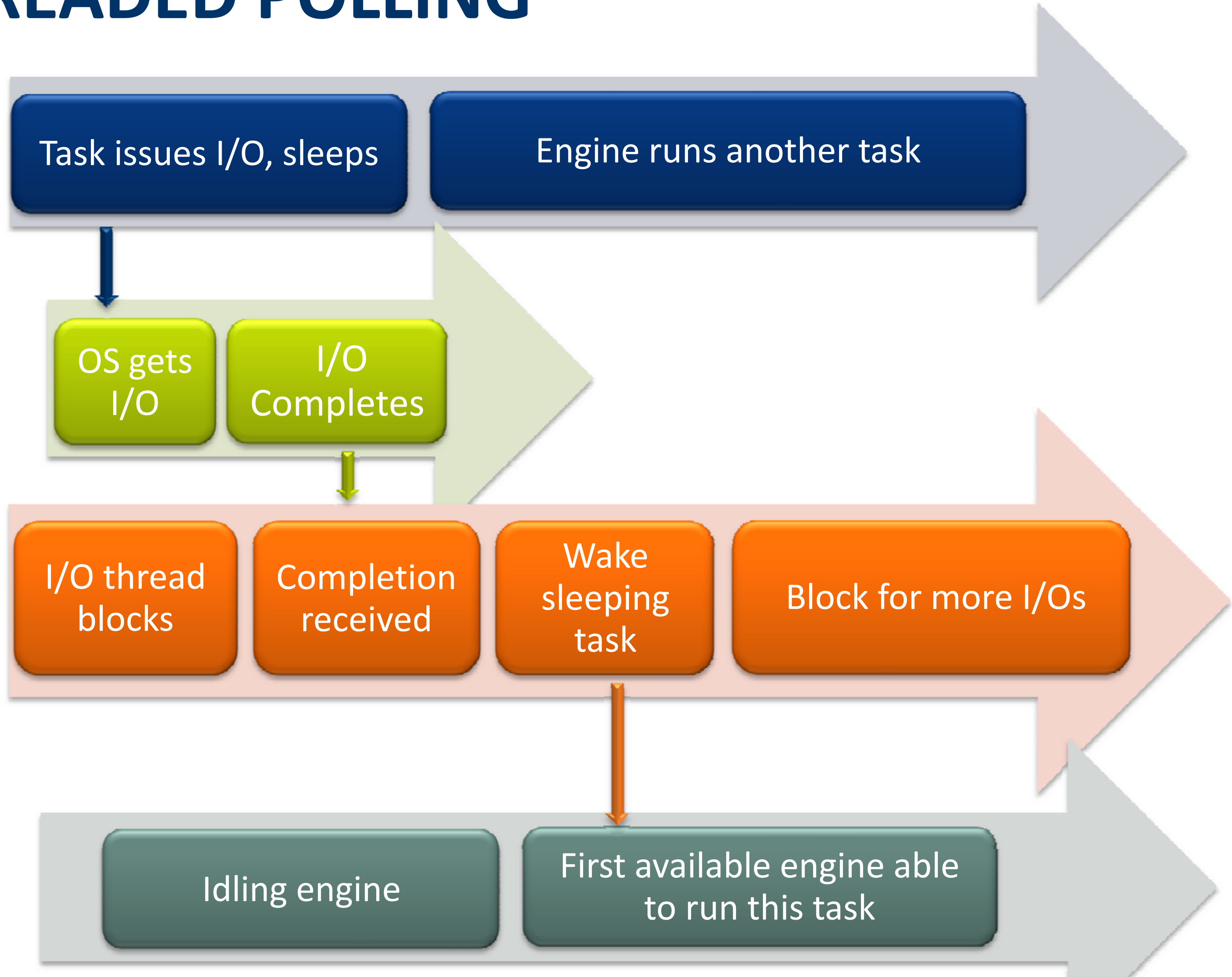
- Dedicated I/O threads handle polling
- Threads block in the O/S until an I/O completes
- Completion is done in the thread, in parallel with engines
- Any available engine can schedule the task

INLINE I/O POLLING (PROCESS KERNEL)

How the process kernel handles I/O. We call this inline polling.



THREADED POLLING



SYBASE | An SAP Company

NETWORK I/O

Process Mode

- Connection owned by an engine
- All I/O for a connection must be done owning engine
- Task on non-net engine must post the send to be done by network service task
- Engine polls for completion, does actual send / receive

Threaded Mode

- No connection ownership, network I/O affinity
- Tasks can directly send regardless of engine
- Network task in native thread polls for completions
- Network task does receive directly, wakes task
- Improves read-ahead

DISK I/O

Process Mode

- I/O issued in task (SPID) context
- Engine periodically polls for completion
- Engine which issued the I/O must complete the I/O
- Engine will never sleep while disk i/o is outstanding

Threaded Mode

- I/O still issued in task (SPID) context
- Disk task in native thread handles completion
- No engine polling required
- Engines can sleep while disk i/o is outstanding

CT-LIB I/O OF SYSTEM SERVICES

Used by Rep Agent and CIS
(This is not the same as I/O to CT-Lib clients)

Process Mode

- SPIDS doing ct-lib I/O are bound to the engine that initiated the connection
- CT-Lib I/O is owned by the engine that issued it
- Deferred async programming model
- Engines periodically call ct_poll()

Threaded Mode

- No engine affinity required
- Full async programming model
- Engines do not call ct_poll()
- Ct-lib's async thread completes the I/O and wakes the ASE task

LEVERAGING LATEST APIS

I/O handling rework includes moving to more efficient APIs

Process Mode

- Uses select() or poll()
- Inefficient for large numbers of connections
- Some sites have reported heavy network spinlock contention with many engines and many clients

Threaded Mode

- Uses modern platform specific APIs
- Efficient for large numbers of connections
- Event model avoids costly code path and allows spinlocks to be eliminated

IMPORTANT POINTS OF THREADED POLLING

Some new things to keep in mind…

I/O Threads Consume CPU Cycles

- With heavy I/O load
- On CPUs with low single thread performance

Factor I/O Thread CPU Consumption into Configuration

- Engines do less work, so you may need fewer…can offset I/O CPU load

I/O Capacity

- Scales in process mode as you add engines
- Requires different scaling in threaded mode

MULTIPLE I/O CONTROLLERS

Additional I/O threads can be configured on Linux, Solaris, AIX, and HP (network only)

Do I Need This?

- Single thread on Xeon / Linux handles 200K+ I/Os a second
- See sysmon (later in this section)

Configuration

- "number of network tasks" & "number of disk tasks"
- Dynamic increase (but doesn't rebalance), reboot to shrink

Watch Out for Sun's T-Series

- ~20k net I/Os per second per thread on T5440
- Platform needed lots of OS tuning
- New T4 CPUs supposedly fixed single-threaded performance

HP-UX I/O ENHANCEMENTS

Improved I/O performance and easier management on HP-UX

Direct & Concurrent I/O

- Direct I/O now supported for JFS files
- Concurrent I/O supported for Online JFS

Async I/O

- Async I/O is now supported for file system devices
- Must configure “enable hp posix async i/o”
- Raw and file devices can coexist

REDUCED CONTENTION VIA ATOMIC OPS

Internal contention reduced using hardware provided atomic instructions

Atomic Counters

- Fetch and add based
- Several global counters are now spinlock-free

Spinlock Elimination

- Internal date/time treated at 64-bit atomic, spinlock eliminated

SPARC Optimization

- Spinlock assembly code has optimization for CMT processors

Compare and Swap

- Object manager spinlock reduced via CAS-based keep count
- Database timestamp

Concurrent Reader Locks

- Compare and swap based
- Significantly reduces contention on rarely modified internal structures

AGENDA

- Architectural Introduction
- Workloads
- Configuration and Tuning
- Interpreting sp_sysmon

WORKLOADS & PERFORMANCE

This is a kernel project. Workloads that didn't spend time in the kernel won't see a difference.

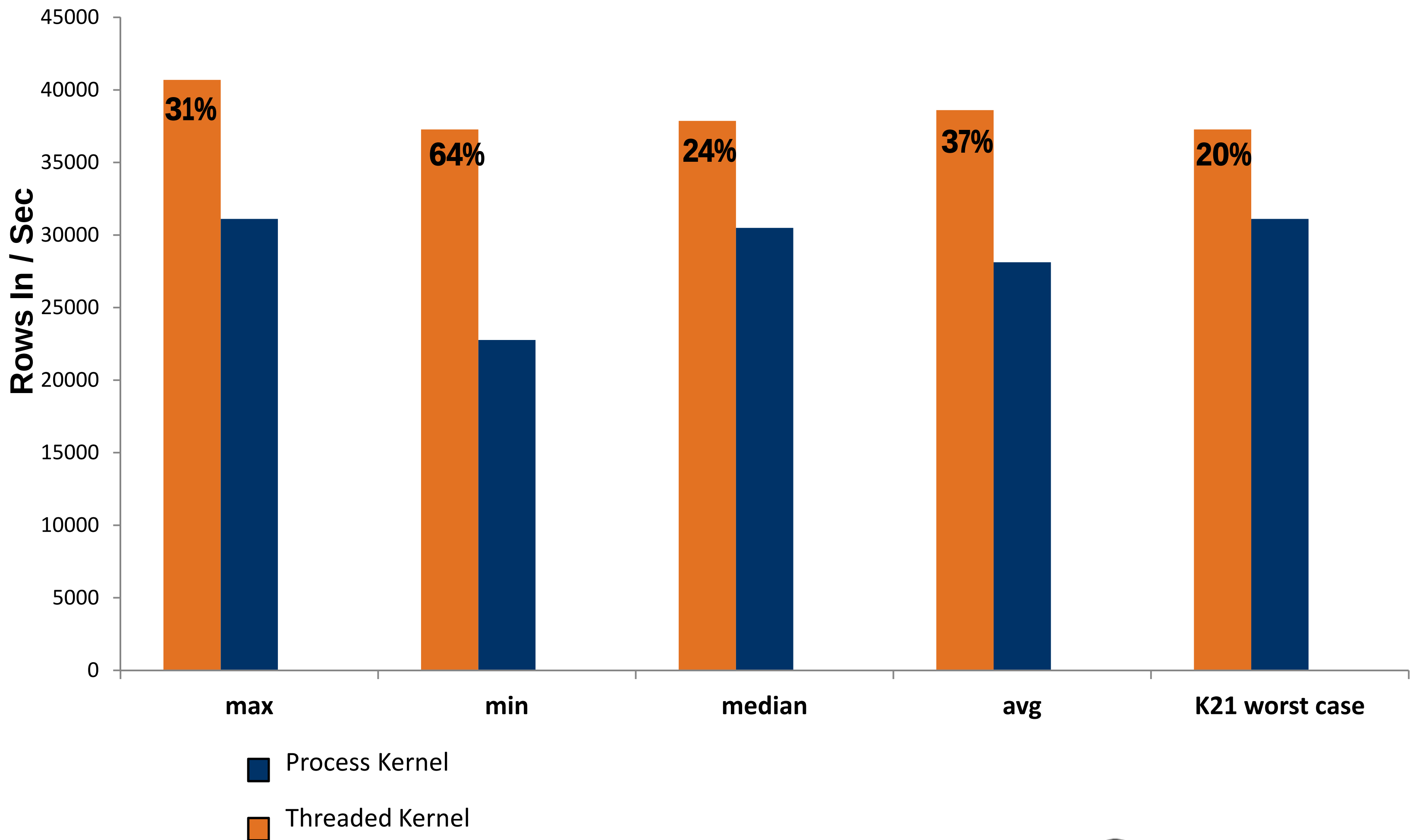
Not So Useful For…	Especially Good For…	Look in Sysmon For…
• Workloads that didn't hit kernel obstacles • Individual queries in small multi-user environments	• Mixed CPU and I/O bound workloads • Bursts of connections • Rep Agent and CIS (proxy table) users	• High I/O busy • Unbalanced engine CPU or network usage • Lots of context switches due to network send

SYBASE | An SAP Company

YOUR MILEAGE WILL VARY

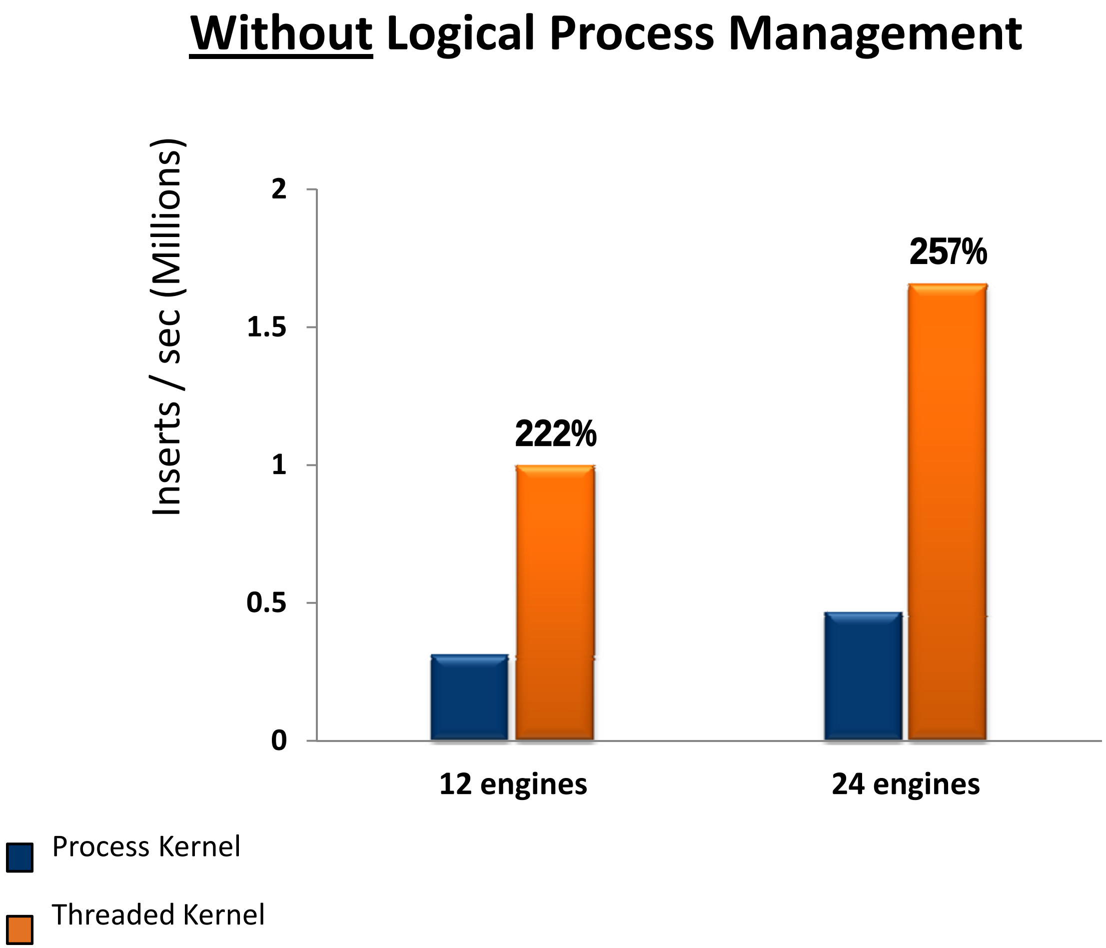
The following slides show results of various performance tests and highlight the benefits of the threaded kernel. Not all workloads will see similar benefits and your mileage will vary.

BCP IN CUSTOMER BENCHMARK



RAP BENCHMARK: "OUT OF THE BOX"

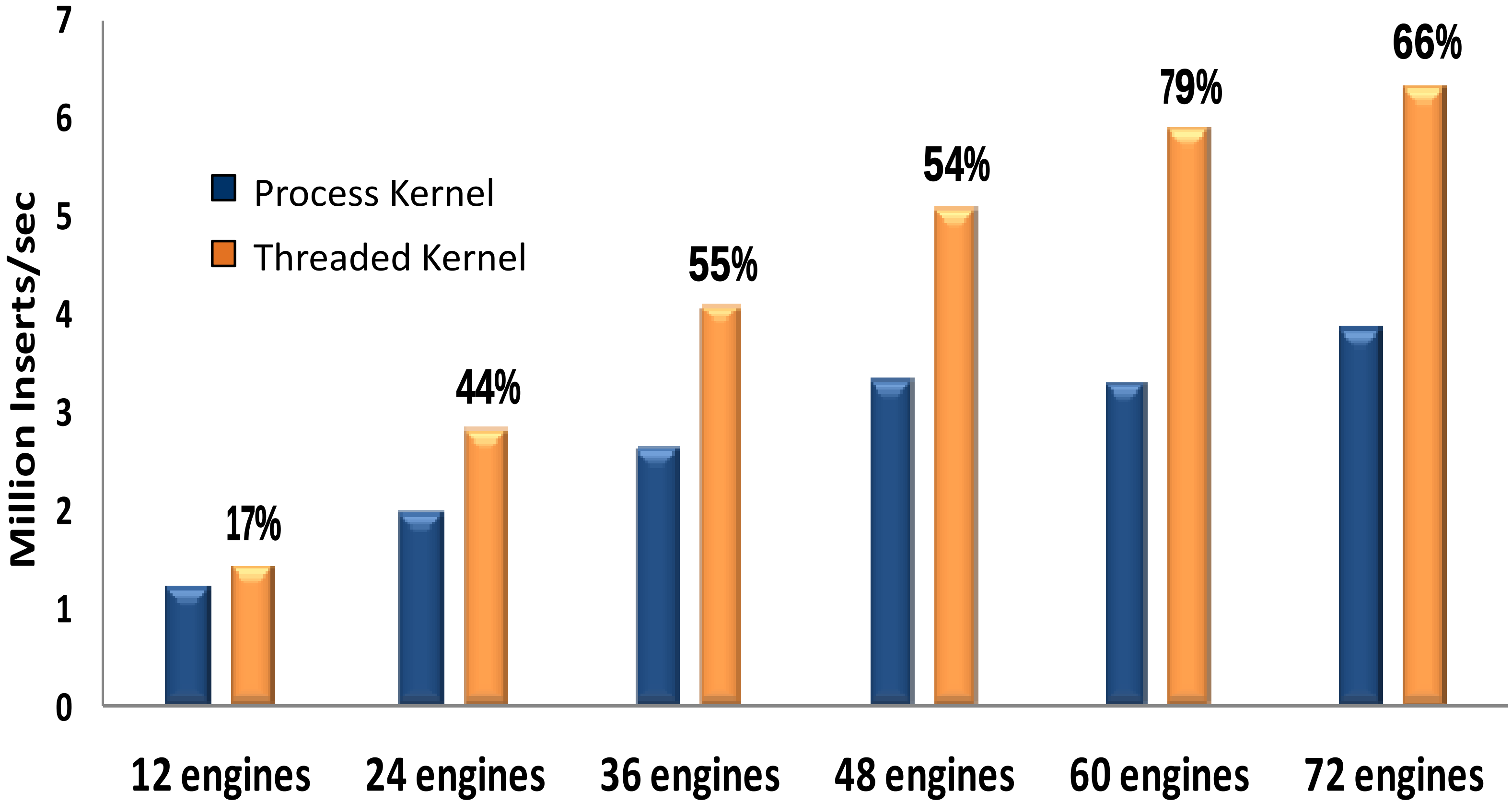
Mixes inserts, deletes, and queries



RAP makes extensive use of engine groups, execution classes, and dynamic listeners to work around limitations in the kernel. These are no longer necessary in the threaded kernel

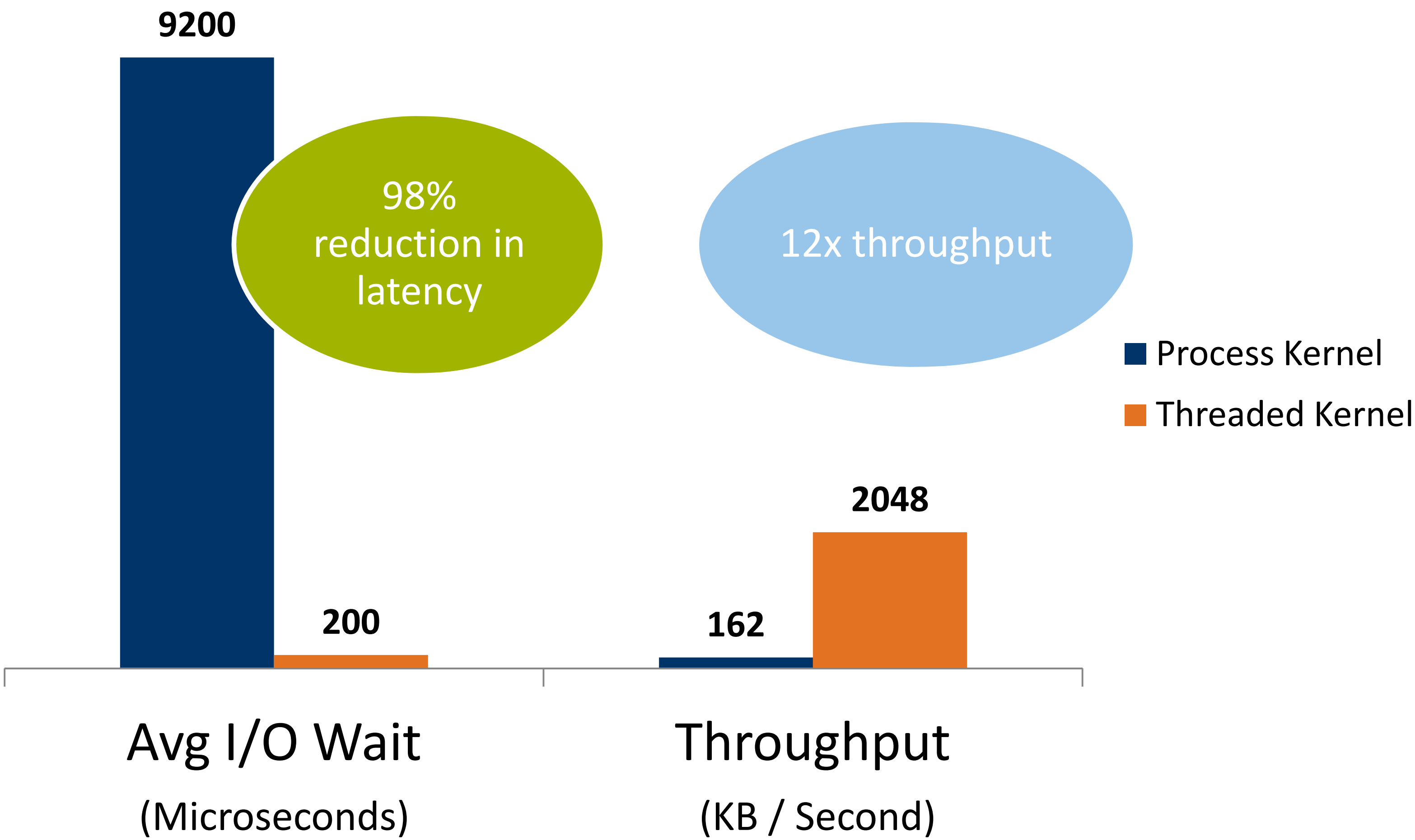
RAP BENCHMARK SCALING

Here the process kernel is helped by engine groups and dynamic listeners.
No such tuning is needed for the threaded kernel.

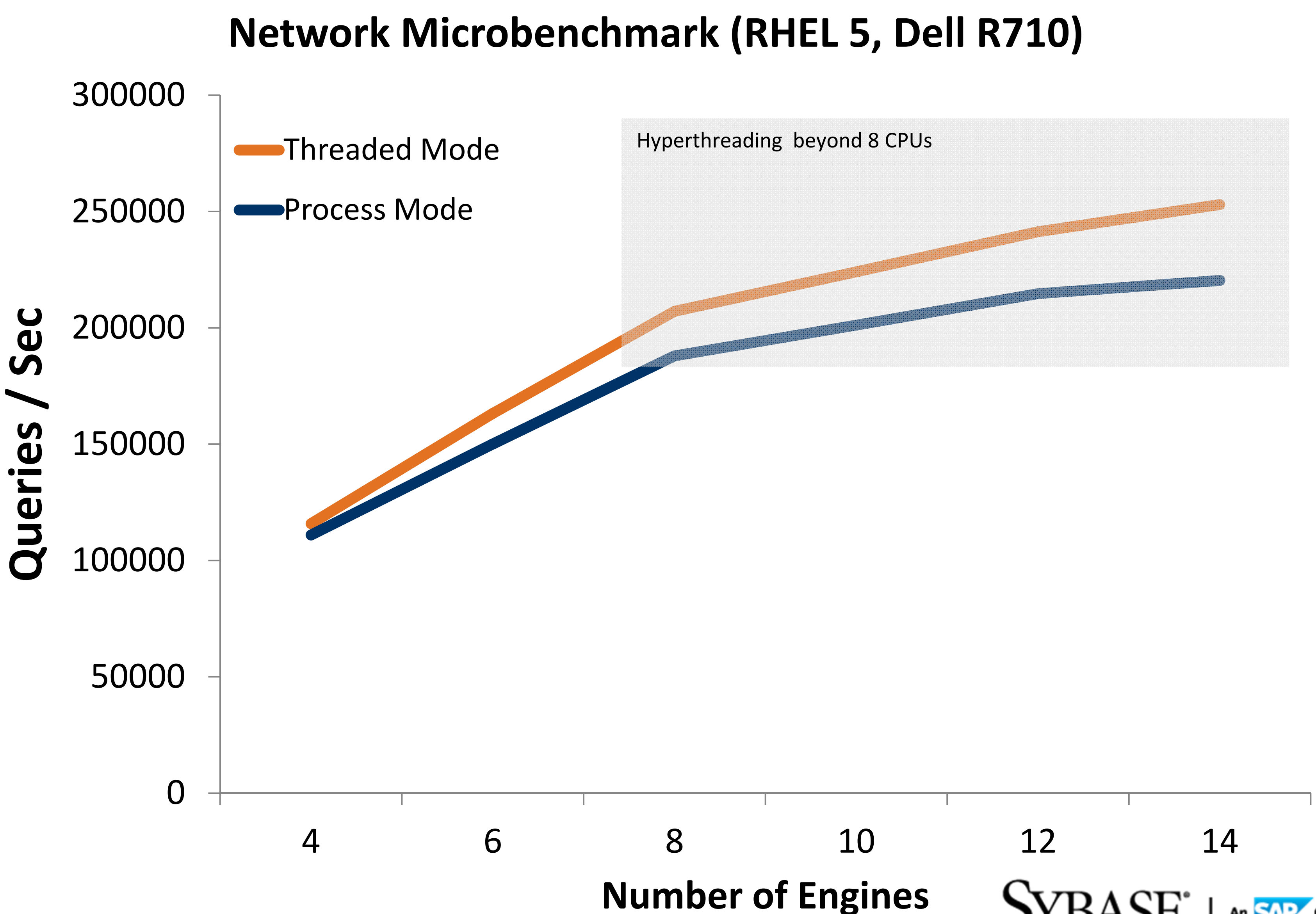


REP AGENT PERFORMANCE

Simple internal test using 15.7 multiple rep agents

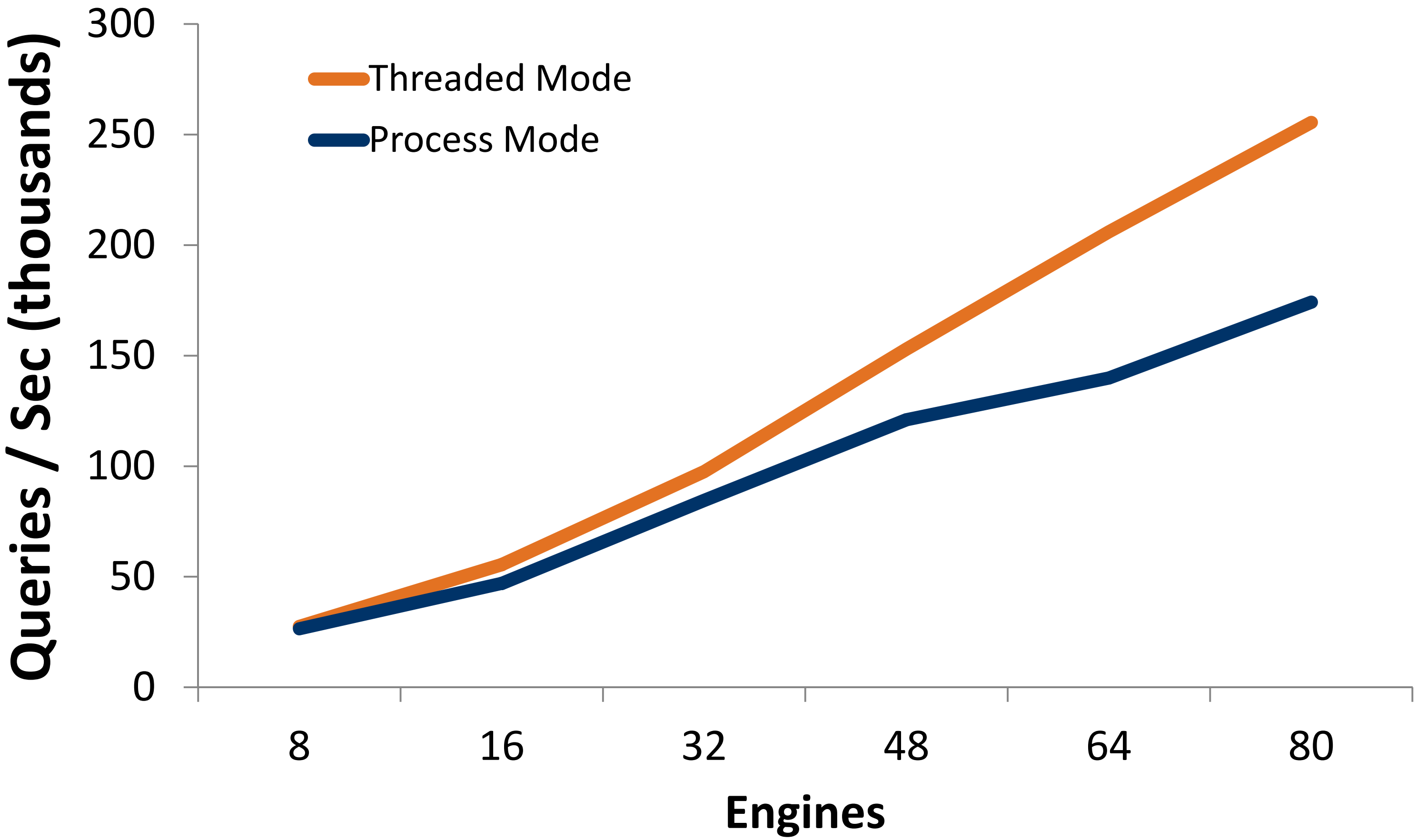


NETWORK SCALING ON NEHALEM-EP



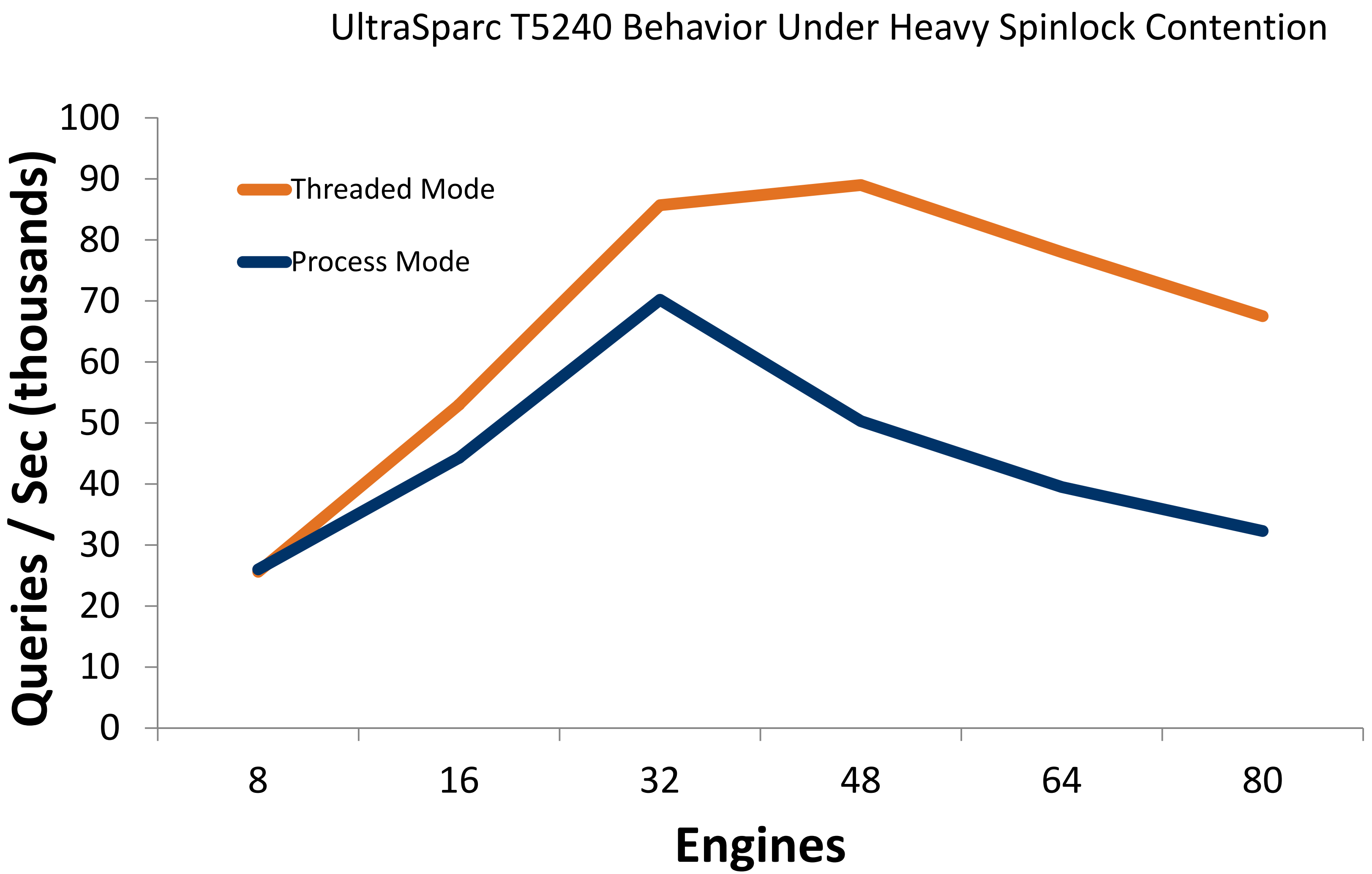
SYBASE | An SAP Company

NETWORK SCALING ON ULTRASPARC T5440



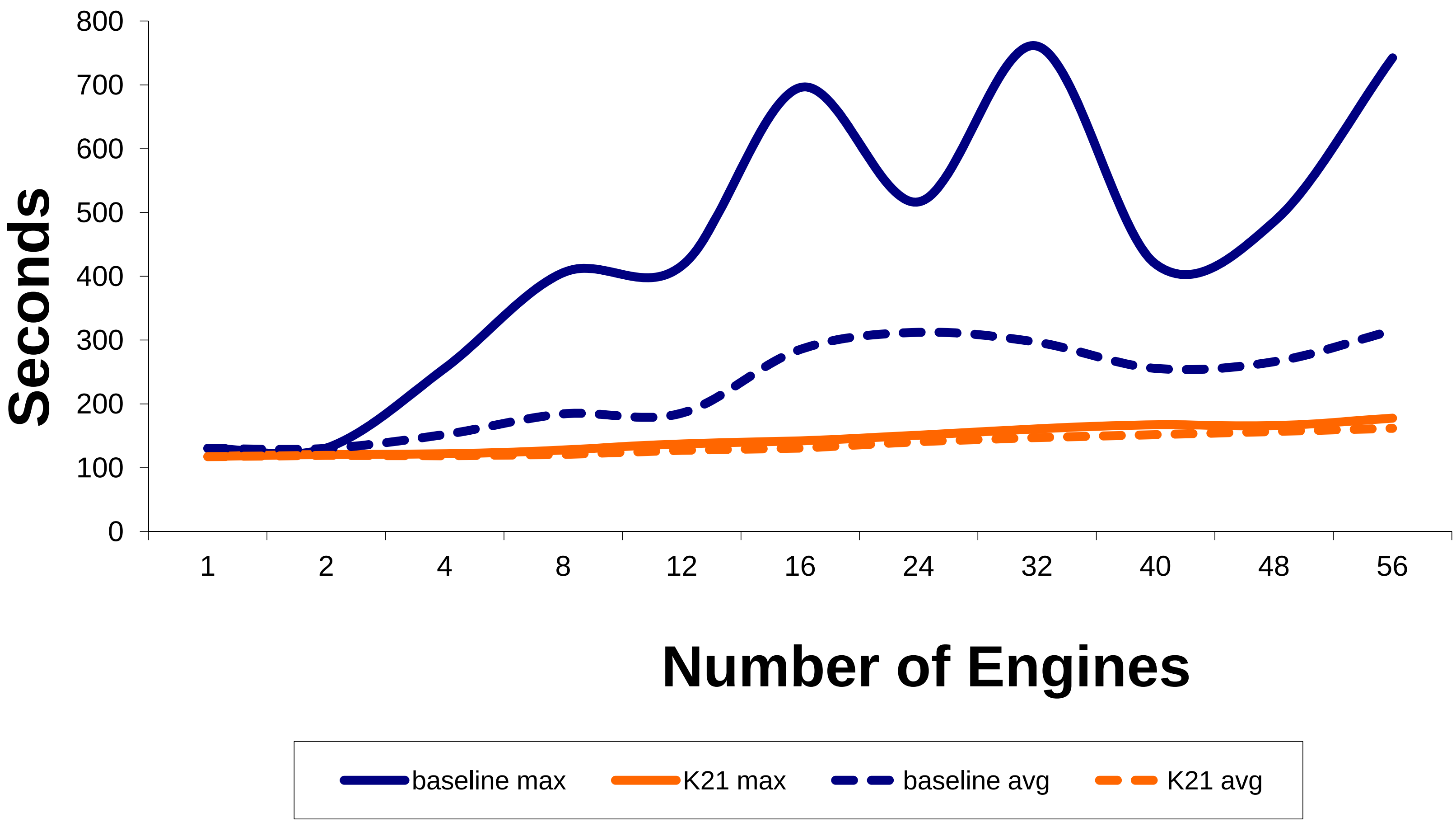
SYBASE | An SAP Company

T-SERIES SPINLOCK IMPROVEMENTS



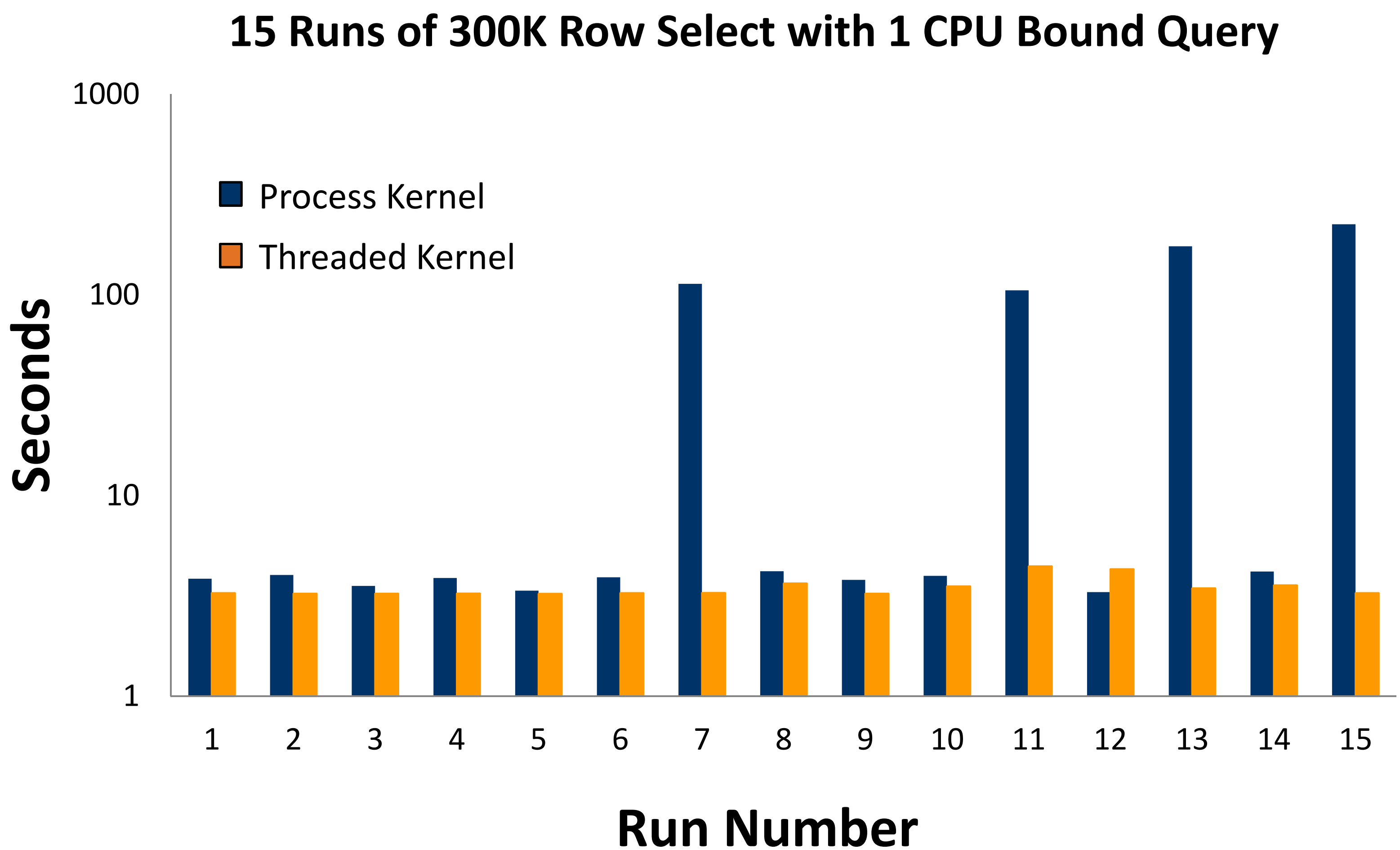
SYBASE | An SAP Company

FSI SELECT / CPU MIX



SYBASE | An SAP Company

CONSISTENT PERFORMANCE: SELECT SPIKE



AGENDA

Architectural Introduction

Workloads

Configuration and Tuning

Interpreting sp_sysmon

BEFORE YOU BOOT: FILE DESCRIPTORS

Process Kernel

- Each engine adds to ASE's descriptor pool
- Total descriptors = per-process limit * engines

Threaded Kernel

- Total descriptors = per-process limit
- You may need to adjust your shell limits for large user configurations

KERNEL RESOURCE MEMORY

sp_configure "kernel resource memory"
Applies to both kernel modes

Controls the amount of memory available for kernel allocations

Rules of thumb:

- <= 8 engines: 1 page per two user connections
- > 8 engines: default + 1 page per user connection

Check utilization with sp_monitorconfig

CONFIGURING ENGINES

Process Kernel

- "max online engines" is a static limit
- "number of engines at startup" is the boot number
- Online and offline via sp_engine
- Single "engine space"

Threaded Kernel

- "max online engines" is still a static limit
- "number of engines at startup" **is ignored!**
- Engines based on explicit thread pool sizes
- "alter thread pool" to online & offline

CONFIGURATION SEQUENCE

Process Kernel

Sp_configure "max online engines", 8

Sp_configure "number of engines at startup", 8

Reboot

Threaded Kernel

Sp_configure "max online engines", 8

Reboot

alter thread pool syb_default_pool with thread count = 8

HOW MANY ENGINES DO I HAVE?

Total engines = syb_default_pool thread count + user pool thread counts

Select count(*) from sysengines still works

Can't be more than "max online engines"

UPGRADING FROM OLDER KERNELS

No actual upgrade is needed

Thread pool configuration is missing from config file → ASE assumes you are upgrading from an older configuration → Default thread pool layout created → syb_default_pool sized based on “number of engines at startup”

THREAD POOL COMMANDS

create thread pool	alter thread pool	drop thread pool	sp_helpthread
• Creates a new pool of engines • New engines are created – they aren't taken from syb_default_pool	• Modify an existing thread pool • This is the way to reconfigure the number of engines	• Removes a user created thread pool • Engines are taken offline and destroyed	• Displays current thread pool configuration • Wrapper around monThreadPool

SYBASE | An SAP Company

THREAD POOL ATTRIBUTES

Attributes	Examples
• Name • Description • Thread Count (size) • Idle Timeout	• alter thread pool syb_default_pool with thread count = 32 • create thread pool sales_pool with thread count = 4, idle timeout = 500, description = “thread pool for sales users”

THREAD POOLS AND THE CONFIG FILE

Thread pools are stored in the configuration file similar to data caches

```
[Thread Pool:syb_default_pool]
  number of threads = 32

[Thread Pool:sales_pool]
  number of threads = 4
  description = thread pool for sales users
  idle timeout = 500

[Thread Pool:syb_blocking_pool]
  number of threads = 4
```

USER THREAD POOLS

User thread pools replace engine groups in threaded mode

Allow you to divide engines into different pools

Use execution classes to assign tasks to thread pools

Better resource isolation and scaling than engine groups

ENGINE GROUPS VS THREAD POOLS

Engine Group

- Used to restrict app to a specific set of engines
- Engines can still run other tasks
- Engines will search runnable queues of all other engines
- Single scheduler search space

Thread Pool

- Used to divide resources
- Engines in a thread pool can only run tasks assigned to that thread pool
- Engines will not search runnable queues outside of that thread pool
- Scheduler search space is limited to the thread pool

IDLE TIMEOUT

Idle timeout replaces “runnable process search count”
Both control how engines spin before sleeping

RPSC

- Set via sp_configure
- Server-wide, applies to all engines
- Is a loop count… implies dramatically different search time across machines

Idle Timeout

- Set via “alter thread pool”
- Attribute of a thread pool, can mix values in a server
- Is a time quantum in microseconds, consistent across machines

TUNING IDLE TIMEOUT

Threaded Kernel

- 100 microsecond default
- Dynamic & easily changed at runtime
- Too low == added latency due to engine sleep / wake
- Too high == wasted CPU cycles
- Value of 0 means never spin, sleep immediately
- Value of -1 means never sleep, always spin
- Use sysmon, discussed later

TUNING TIP: DON'T OVER CONFIGURE

Threaded Kernel

Threaded kernel is less forgiving if you over configure engines (threads) relative to available CPU

Be sure to leave at least one "CPU" for I/O threads…maybe more

Don't configure 16 engines on an 8 core, hyperthreaded system

Useful sysmon enhancements to help you out

TUNING TIP: THREAD AFFINITY ON X86-64

Both Process and Threaded Kernels

Try to keep dataserver on as few sockets as possible

Cross socket migration is not CPU cache friendly

numactl on linux is your friend…use it if you can

Access to "farthest" memory 2.5X slower than "nearest" memory

AGENDA

- Architectural Introduction
- Workloads
- Configuration and Tuning
- Interpreting sp_sysmon

INTRO TO SYSMON KERNEL CHANGES

New sysmon kernel section for threaded mode

- You need mon_role
- You will get warning messages if “enable monitoring” set to 0, but it works fine

Visibility to OS data

- Reports OS utilization + traditional engine utilization
- CPU consumption of non-engine threads
- Page faults and OS context switching

ENGINE TICKS

Classic engine utilization
Output grouped by thread pool

```
Engine Utilization (Tick %)   User Busy   System Busy   I/O Busy       Idle
 -------------------------  ------------  ------------  ----------  ----------
 ThreadPool : DavePool
  Engine 3                         0.0 %         0.0 %       0.0 %     100.0 %

 ThreadPool : syb_default_pool
  Engine 0                        98.7 %         1.3 %       0.0 %       0.0 %
  Engine 1                        98.7 %         1.3 %       0.0 %       0.0 %
  Engine 2                        98.7 %         1.3 %       0.0 %       0.0 %
 -------------------------  ------------  ------------  ----------  ----------
 Pool Summary         Total      296.0 %         4.0 %       0.0 %       0.0 %
                    Average       98.7 %         1.3 %       0.0 %       0.0 %

 -------------------------  ------------  ------------  ----------  ----------
 Server Summary       Total      296.0 %         4.0 %       0.0 %     100.0 %
                    Average       74.0 %         1.0 %       0.0 %      25.0 %
```


SYBASE | An SAP Company

RUN QUEUE "DEPTH"

Always shows 1, 5, and 15 minute avg regardless of sysmon interval
High numbers here likely indicate more engines are needed.

```
Average Runnable Tasks              1 min         5 min      15 min  % of total
  -------------------------  ------------  ------------  ----------  ----------
  ThreadPool : syb_default_pool
   Global Queue                       0.0           0.0         0.0       0.0 %
   Engine 0                           2.1           0.8         0.4      31.2 %
   Engine 1                           2.1           0.8         0.5      30.8 %
   Engine 2                           2.6           0.9         0.5      38.0 %
  -------------------------  ------------  ------------  ----------
  Pool Summary          Total         6.8           2.5         1.4
                      Average         1.7           0.6         0.3
```

CPU YIELDS

Full sleeps are good.
High % of interrupted sleeps may indicate idle timeout is too low

```
CPU Yields by Engine             per sec       per xact        count  % of total
  -------------------------  ------------  ------------  ----------  ----------
  ThreadPool : DavePool
   Engine 3
      Full Sleeps                    22.7          61.9         681      98.6 %
      Interrupted Sleeps              0.3           0.9          10       1.4 %

  ThreadPool : syb_default_pool
   Engine 0
      Full Sleeps                     0.1           0.2           2       0.5 %
      Interrupted Sleeps              2.0           5.5          60      16.5 %
   Engine 1
      Full Sleeps                     0.1           0.2           2       0.5 %
      Interrupted Sleeps              4.3          11.8         130      35.7 %
   Engine 2
      Full Sleeps                     0.1           0.2           2       0.5 %
      Interrupted Sleeps              5.6          15.3         168      46.2 %
  -------------------------  ------------  ------------  ----------
  Pool Summary                       12.1          33.1         364

  -------------------------  ------------  ------------  ----------
```


SYBASE | An SAP Company

THREAD UTILIZATION

Shows OS user / sys time for each thread, engine and non-engine
Only threads with > 0% time are displayed

```
 Thread Utilization (OS %)      User Busy   System Busy        Idle
  ------------------------  ------------  ------------  ----------
  ThreadPool : syb_default_pool
   Thread 1     (Engine 0)        67.2 %        10.7 %      22.1 %
   Thread 2     (Engine 1)        50.4 %         8.9 %      40.7 %
   Thread 3     (Engine 2)        66.8 %        12.6 %      20.7 %
  ------------------------  ------------  ------------  ----------
  Pool Summary       Total       184.4 %        32.1 %      83.5 %
                   Average        61.5 %        10.7 %      27.8 %

  ThreadPool : syb_system_pool
   Thread 8    (NetController)   27.9 %        14.1 %      58.1 %
------------------------  ------------  ------------  ----------
  Pool Summary       Total        27.9 %        14.1 %     358.1 %
                   Average         7.0 %         3.5 %      89.5 %

  ------------------------  ------------  ------------  ----------
  Server Summary     Total       212.2 %        46.2 %     641.6 %
                   Average        23.6 %         5.1 %      71.3 %
```

PAGE FAULTS

Look out for major faults…those are bad and indicate a memory shortage on the host. Not displayed on Windows.

```
Page Faults at OS                per sec      per xact       count  % of total
  -------------------------  ------------  ------------  ----------  ----------
   Minor Faults                       4.8           4.8          48     100.0 %
   Major Faults                       0.0           0.0           0       0.0 %
  -------------------------  ------------  ------------  ----------  ----------
   Total Page Faults                  4.8           4.8          48     100.0 %
```

CONTEXT SWITCHES

High % of non-voluntary likely means the CPUs are over subscribed.
Not shown on Windows. Linux requires RHEL 6 / SLES 11

```
Context Switches at OS            per sec       per xact       count  % of total
  -------------------------  ------------  ------------  ----------  ----------
  ThreadPool : syb_blocking_pool
   Voluntary                          0.0           0.0           0       0.0 %
   Non-Voluntary                      0.0           0.0           0       0.0 %
  ThreadPool : syb_default_pool
   Voluntary                         43.6         130.7        1307      56.3 %
   Non-Voluntary                      0.2           0.6           6       0.3 %
  ThreadPool : syb_system_pool
   Voluntary                         33.7         101.0        1010      43.5 %
   Non-Voluntary                      0.0           0.0           0       0.0 %
  -------------------------  ------------  ------------  ----------  ----------
   Total Context Switches            77.4         232.3        2323     100.0 %
```


SYBASE | An SAP Company

I/O CONTROLLERS

Replaced "disk checks" and "network checks" in process sysmon
Look out for "Polls Returning Max Events" or high Events Per Poll

```
DiskController Activity          per sec      per xact       count  % of total
 -------------------------  ------------  ------------  ----------  ----------
  Polls                              7.7          21.0         231       n/a
  Polls Returning Events             0.1           0.4           4       1.7 %
  Polls Returning Max Events         0.0           0.0           0       0.0 %
  Total Events                       0.1           0.4           4       n/a
  Events Per Poll                    n/a           n/a       0.017       n/a

 NetController Activity           per sec      per xact       count  % of total
 -------------------------  ------------  ------------  ----------  ----------
  Polls                         110680.4      301855.7     3320413       n/a
  Polls Returning Events         26880.5       73310.5      806415      24.3 %
  Polls Returning Max Events         0.0           0.0           0       0.0 %
  Total Events                   32181.4       87767.5      965442       n/a
  Events Per Poll                    n/a           n/a       0.291       n/a
```

BLOCKING CALLS

These are requests to syb_blocking_pool.
Consider increasing the pool size if Queued Requests or Wait Time > 0.

```
Blocking Call Activity          per sec      per xact       count  % of total
 -------------------------  ------------  ------------  ----------  ----------
   Serviced Requests                 0.2           0.4           5     100.0 %
   Queued Requests                   0.0           0.0           0       0.0 %
 -------------------------  ------------  ------------  ----------
 Total Requests                      0.2           0.4           5
 Total Wait Time (ms)                n/a           n/a           0
```


SYBASE | An SAP Company

DANGER SIGNS IN SYSMON

OS Utilization vs. Engine Utilization

- OS < Engine, look for CPU starvation
- Engine >> OS, utilization could mean idle timeout is too high

Deep Run Queues

- Lots of runnable tasks generally means > latency
- Consider adding engines
- Adding engines won't help high utilization but shallow run queues

Full Sleeps vs. Interrupted Sleeps

- Most sleeps should be full
- High % of interrupted sleeps means idle timeout is too low

Major page Faults and Non-Voluntary Context Switches

- Resource shortage (memory, CPU) on the host

High I/O Thread Utilization

- CPU > 70%, consider adding another thread
- If higher than engine utilization, then engine is probably starved

I/O Event Density

- Consider adding an I/O task if you get "polls returning max events" <u>or</u> events per poll > 3

SUMMARY

- Hybrid Threaded Kernel – One of the major innovations in 15.7
  - Continue to leverage the best of VSA (internal DBMS threading)
    - Simplicity, Control, Performance
  - Extend VSA with OS threading for responsiveness of system services
- Many environments will benefit:
  - Environments with heavy Rep Server or CIS usage
  - Mixed CPU & I/O-bounds workloads, especially with disk/network intensive interactions
  - Systems requiring rapid bursts of network logins
  - People using Sun UltraSPARC (T-series) hardware
- Early and extra QA testing of the new kernel by SAP will pay “quality” dividends to Sybase customers

THANK YOU

FOR MORE INFORMATION

WWW.SYBASE.COM/ASEBUILTFORBUSINESS

TYPE YOUR QUESTION IN THE ONLINE Q&A BOX OR CALL

1-866-803-2143 UNITED STATES

001-866-888-0267 MEXICO

0800-777-0476 ARGENTINA

01800-9-156448 COLOMBIA

0800-279-3953 LONDON

1-210-795-1098 ALL OTHER INTERNATIONAL

PASSWORD: SYBASE

PRESS *1 TO ASK A QUESTION

THANK YOU

FOR MORE INFORMATION

WWW.SYBASE.COM/ASEBUILTFORBUSINESS

SYBASE®

An SAP® Company