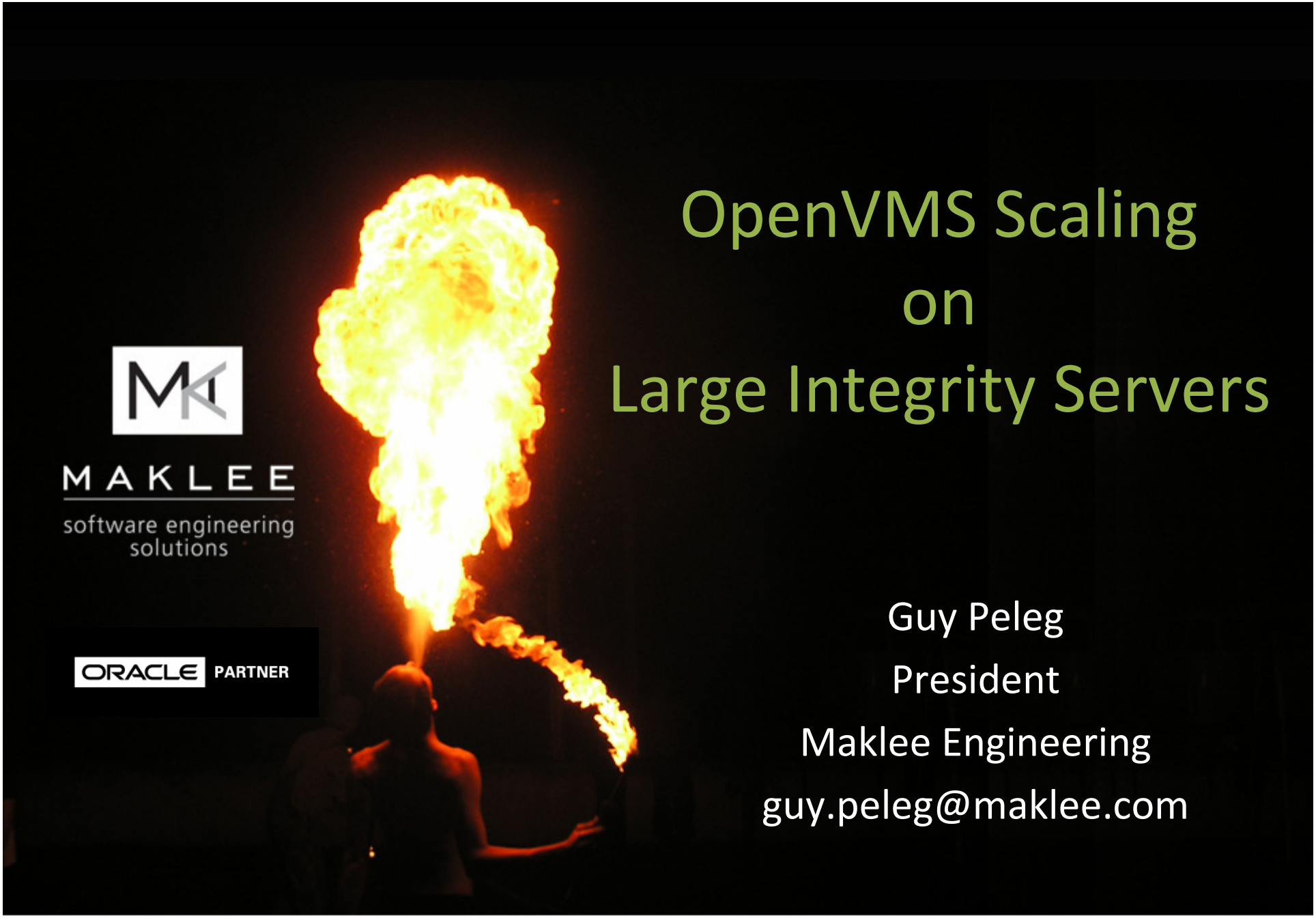


OpenVMS Scaling on Large Integrity Servers

MAKLEE
software engineering solutions

ORACLE PARTNER

Guy Peleg
President
Maklee Engineering
guy.peleg@maklee.com

Who we are

- What is Maklee?
  - US Based consulting firm operating all over the world.
  - Former members of various engineering groups at HP
- Among our customers are:
  - Verizon Wireless, Eli Lilly, AIG financial group, Volvo, M.O.L. America, ConEd, FDNY, France Telecom, IKEA, Navistar, Private Banks in Europe, Frankfurt Airport, ThyssenKrupp Steel, Tel-Aviv Stock Exchange, Hewlett Packard, Dow Jones Company, Bloomberg, NYSE and more…
- We specialize in:
  - Performance Tuning
  - Oracle & Oracle tuning (official Oracle Partner)
  - Platform migration
  - Custom Engineering
- Supported platforms: OpenVMS, HP-UX, Linux, Tru64, Solaris and AIX

MAKLEE

Maklee provides guarantee for success for all projects

What is Scaling?

- ”How well a solution to some problem will work when the size of the problem increases”.
  - Source: “the free dictionary by Farlex”.
- In a perfect world, adding more resources to the computer would allow us to perform more work.
- Doubling the number of CPUs == Doubling the number of transactions?

In most cases the answer is NO !!!

- In some cases, adding more resources lowers throughput.

What is Scaling?

- Is scaling the same as performance?
- What can prevent an application from scaling?
  - Hardware Capacity
  - Latency
  - Contention for shared resource
    - Spinlock
    - Lock
    - Mutex
    - Mailbox
  - Alignment faults

MAKLEE

Integrity Servers

- Small/Medium servers:
  - rx2660
  - rx3600
  - rx6600
  - BL870c
    - 8 cores, 192GB RAM
- Large servers:
  - Rx7640
    - 16 cores, 256GB RAM
  - Rx8640
    - 32 cores, 512GB RAM
  - Superdome (SD16B or SD32B)
    - 128 cores, 2TB RAM

MAKLEE

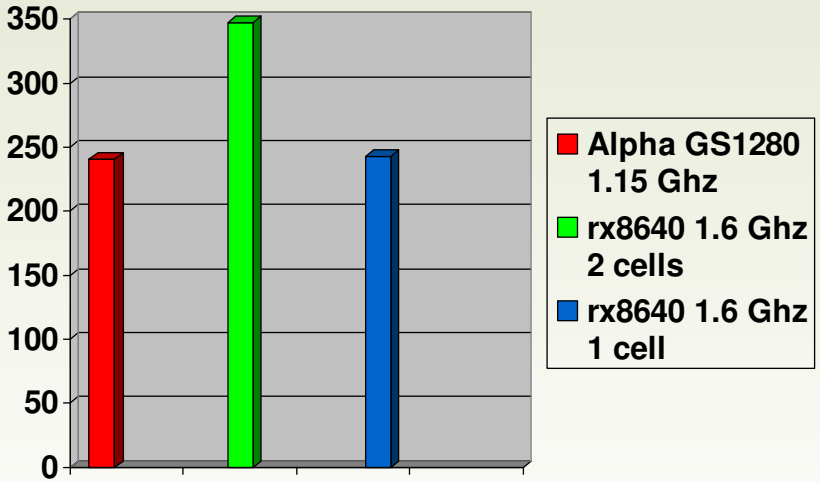
Integrity Servers

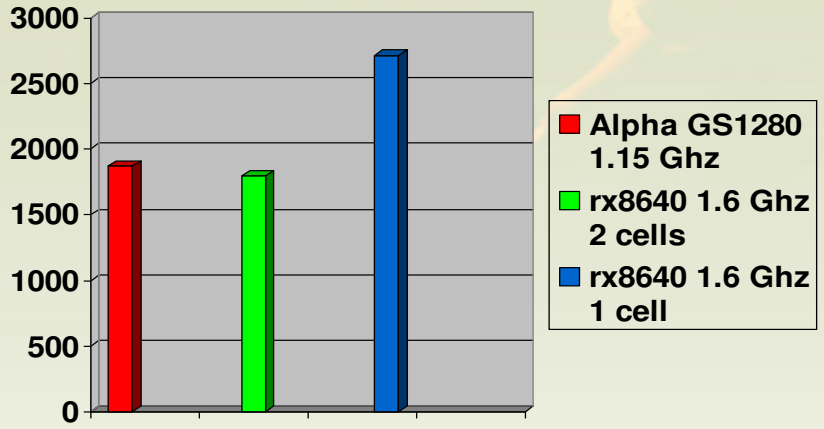
- Memory latency is key for scaling.
  - How long a CPU has to wait before data arrives into the CPU from physical memory.
- Itanium processors are fast and have larger on-chip caches compared to Alpha processors
  - Itanium: 18MB or 24MB per processor
  - Alpha EV7: 1.75MB
- Memory latency is higher than what we are used to on Alpha
  - rx7640/rx8640: 300 nsec to 400 nsec
  - GS1280: less than 200 nsec

MAKLEE

Memory Benchmarks

Memory latency (ns)
less is better





Memory Bandwidth (MB)
More is better

MAKLEE

Integrity Servers

- Memory bandwidth on Integrity outperforms Alpha.
- Large Integrity servers have higher Memory latency than Alphasever GS1280.
  - The engineers of the GS1280 did an outstanding job dealing with latency.
  - Moving from large GS1280 to a large Integrity box may result in performance degradation.

MAKLEE

Superdome NUMA Effect

- Memory latency
  - Local cell ~200 nsec
  - Remote cell ~400 nsec
- Memory throughput
  - Memory/cell 7.2 GB/sec
  - Crossbar/cell 6.4 GB/sec
- Cache line
  - 128 Bytes

Cellular System

Cell #0 CPUs 0-7 Memory	Cell #1 CPUs 8-15 Memory
Cell #2 CPUs 16-23 Memory	Cell #3 CPUs 24-31 Memory

MAKLEE

The Golden Rules

- Run your application on the smallest Integrity server that fits your workload.
  - rx6600 and blade BL870c provide outstanding performance, with very low memory latency.
  - On a cellular system, do not turn on extra cell unless you REALLY need it.
- For workloads that do not fit a small system:

A process should be "close" to it's memory

MAKLEE

Spinlocks

- “a **spinlock** is a lock where the thread simply waits in a loop ("spins") repeatedly checking until the lock becomes available. The thread remains active but isn't performing a useful task”
  - Source: Wikipedia
- OpenVMS uses spinlocks to synchronize access of CPUs for shared resources.
- On OpenVMS, Spin time is counted as MP Sync.
- Common OpenVMS Spinlocks:
  - SCHED
  - LCKMGR
  - IOLOCK8
  - MMG

MAKLEE

CPU Utilization

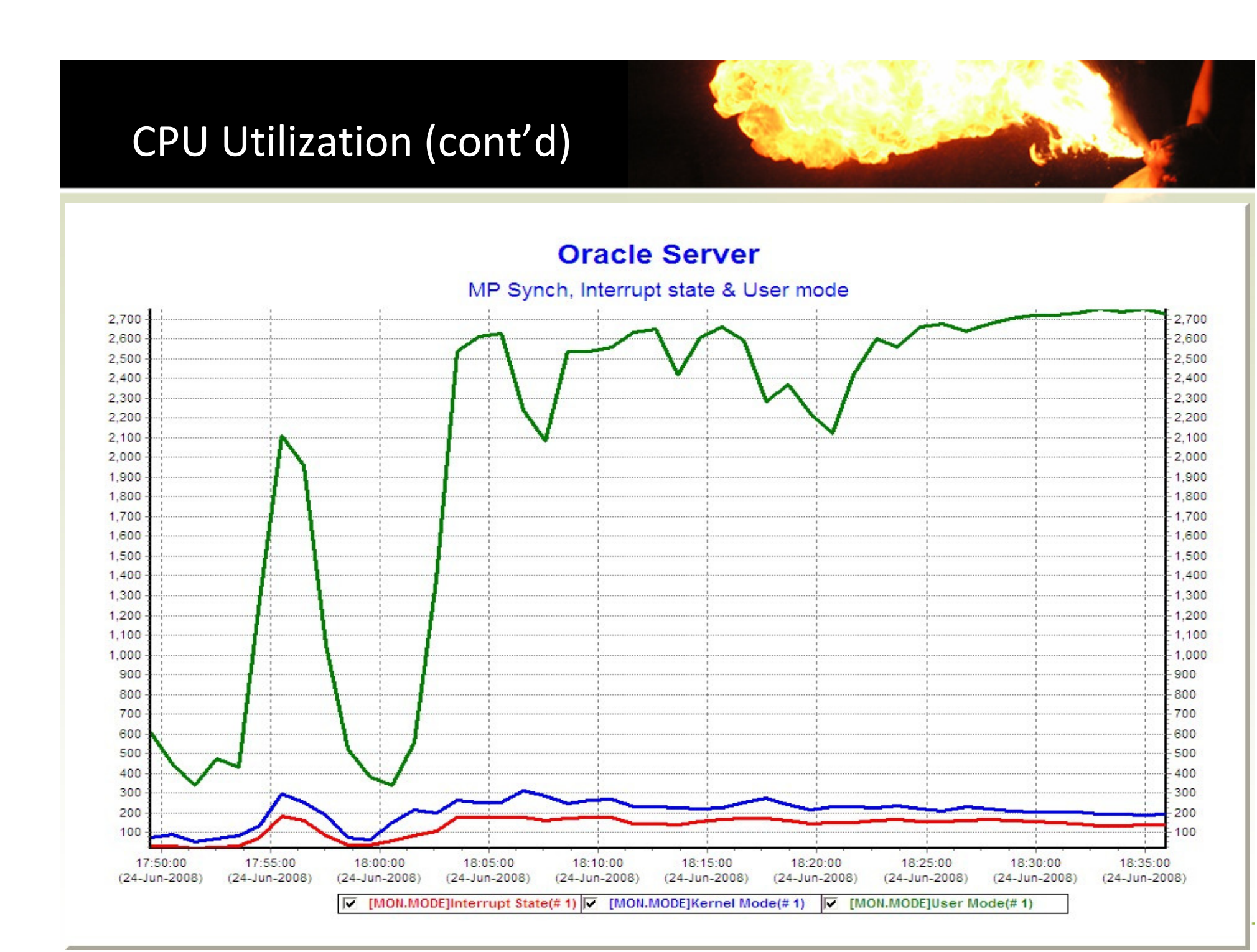
- CPU utilization is like Cholesterol. Can be good or bad.
- Applications should inspire to spend most of their time in user mode.
- OpenVMS modes:
  - User
  - Supervisor
  - Kernel
  - Interrupt
  - Mp Sync
- Bad CPU time
  - Lots MPsync, the CPU is spinning but no useful work is done
  - High kernel mode time
- Good CPU time
  - High user mode time, application is making lots of progress

MPsync

```
                          OpenVMS Monitor Utility
            +-----+       TIME IN PROCESSOR MODES
            | CUR |            on node XYZ
            +-----+       13-JUL-2009 16:28:47.02

Combined for 32 CPUs                    0         800       1600      2400    3200
                                        + - - - - + - - - - + - - - - + - - - - +
 Interrupt State                  101   |a
                                        |         |         |         |         |
 MP Synchronization              2150   |aaaaaaaaaaaaaaaaaaaaaaaaa
                                        |         |         |         |         |
 Kernel Mode                      238   |aa
                                        |         |         |         |         |
 Executive Mode                   456   |aaaaa
                                        |         |         |         |         |
 Supervisor Mode                        |
                                        |         |         |         |         |
 User Mode                        138   |a
                                        |         |         |         |         |
 Compatibility Mode                     |
                                        |         |         |         |         |
 Idle Time                        120   |a
                                        + - - - - + - - - - + - - - - + - - - - +
```

MAKLEE

CPU Utilization (cont'd)



Spinlocks

- Maklee discovered that the physical location (cell) from which memory holding the data structure of a Spinlock is allocated, will impact performance of an application.

- On a cellular system, the physical location of the spinlock may change with every boot.
  - TCP/IP spinlock location may change when TCP/IP is restarted.

- This may impact performance up to 20%

Location, Location, Location

Cell #0	Cell #1
CPUs 0-7 SCHED spinlock	CPUs 8-15 TCPIP Global spinlock TCPIP (BG0)
Cell #2	**Cell #3**
CPUs 16-23 LCKMGR spinlock Dedicated Lock Manager	CPUs 24-31 Dedicated Lock Manager

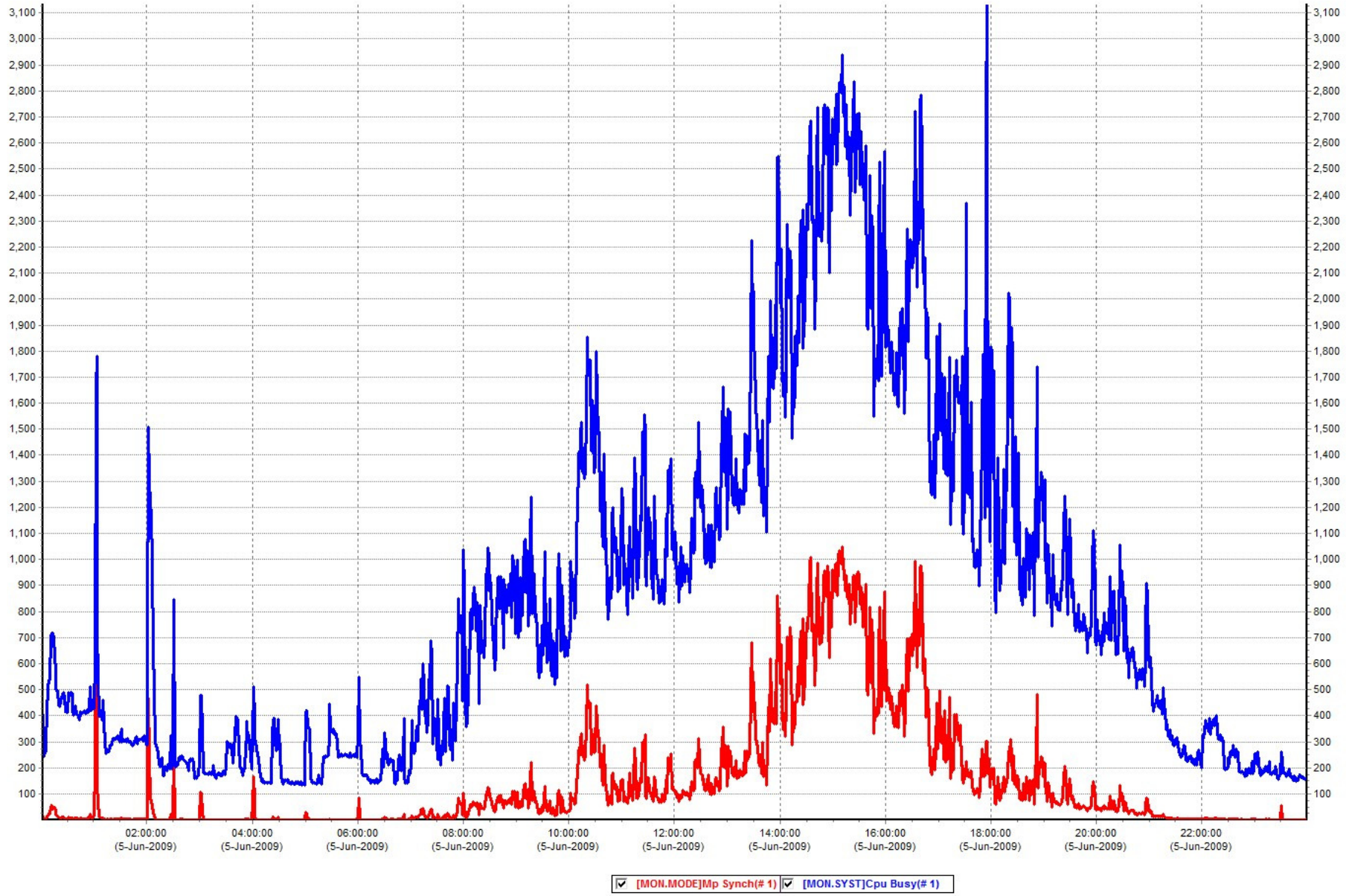
MAKLEE

Example only !

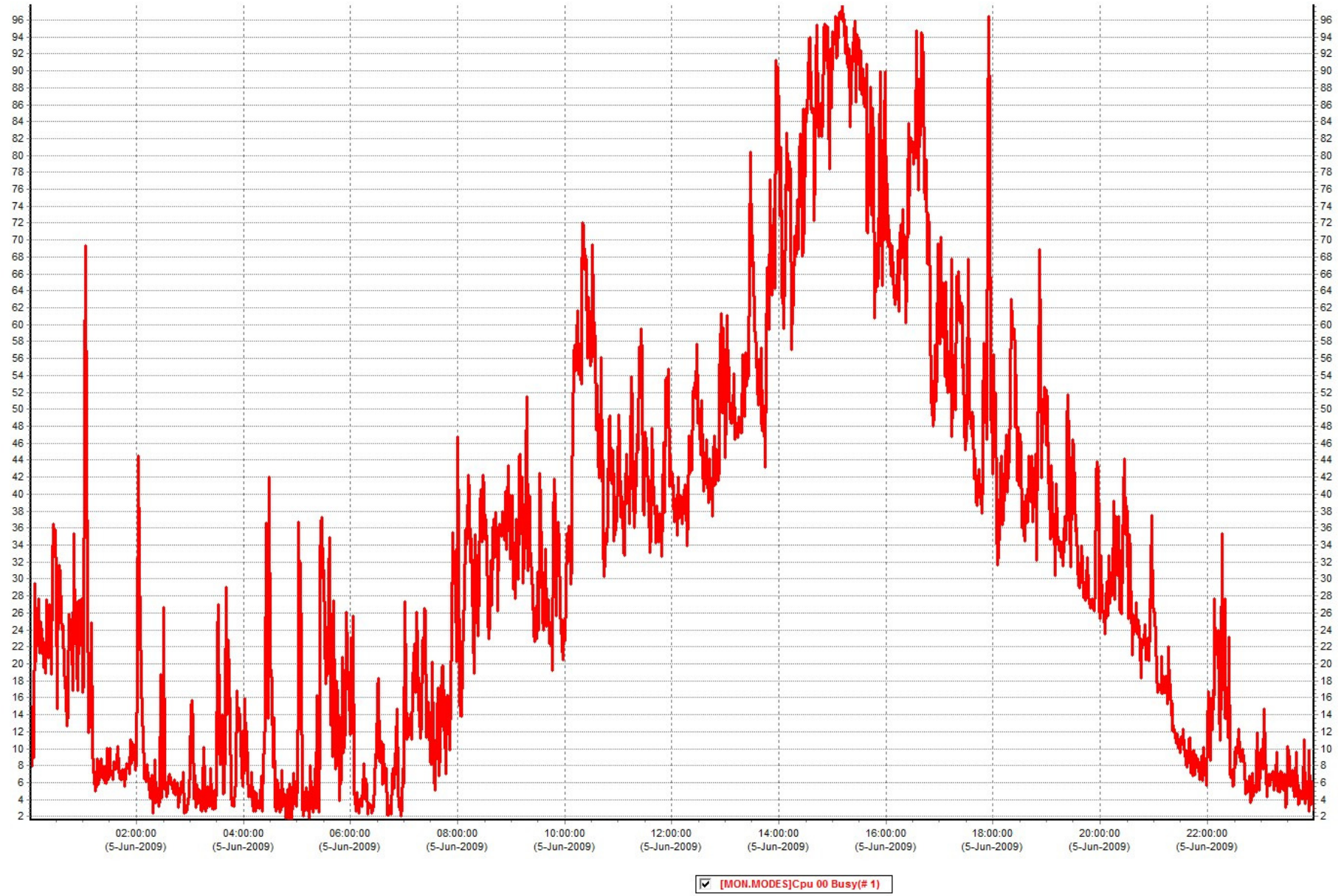
Real life example

- SD32B 1.6GHz / 9MB
- 32 cores
- OpenVMS V8.3-1H1
- During business hours CPU Utilization is close to 100%
- 10 CPUs in MP Sync

CPU Utilization Vs. MP Synch



CPU 0 Utilization



Real life example

- Maklee tuned the system using the following golden rule:

A process should be "close" to it's memory

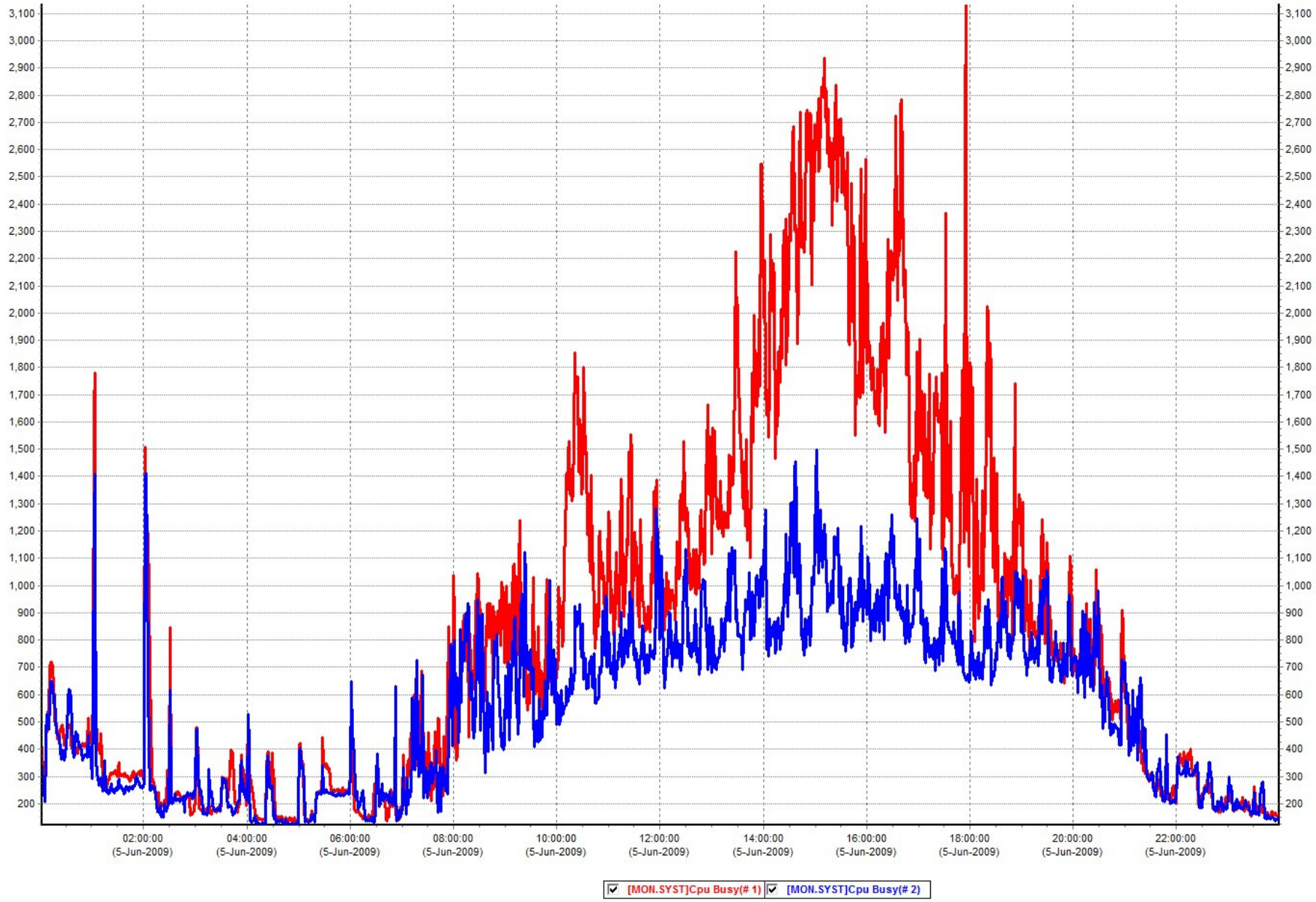
- After tuning, the same workload requires half the computing resources.

- Plenty of room to grow.

MAKLEE

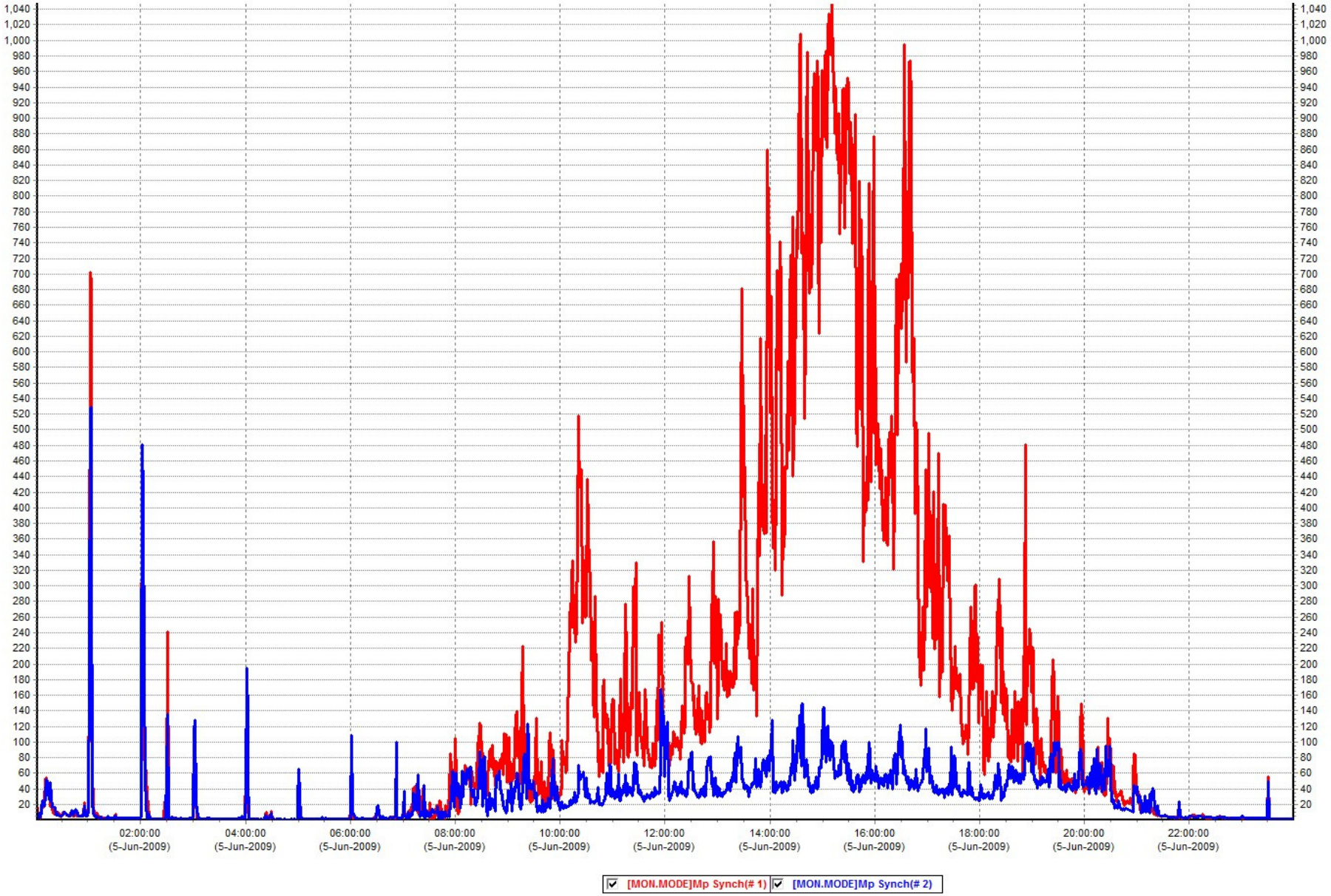
CPU Utilization

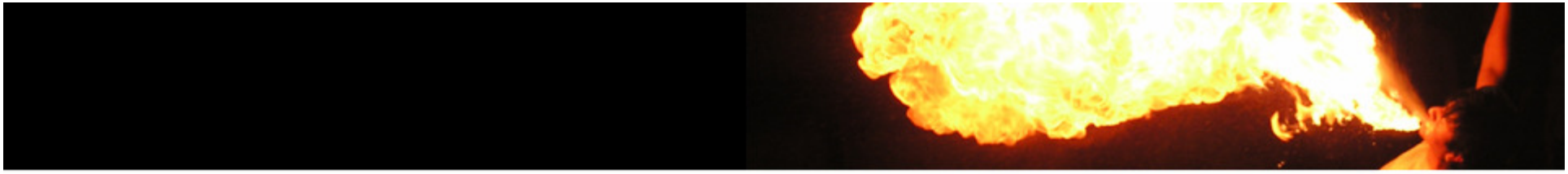
RED=Before BLUE=After



MP Synch

RED=Before BLUE=After





How to improve scaling on large servers?

MAKLEE

OpenVMS

- Current OpenVMS version only supports interleaved memory
  - Cache lines within a page are round-robin interleaved between cells
- No control from which cell memory is allocated from
- Larger pieces of memory evenly distributed across multiple/all cells
  - It doesn't matter from which cell you access the memory
- Smaller chunks (like spinlock structures) allocated from a single cell
  - Causes fairness and access time issues under contention
  - Not necessary persistent across system reboots
- The only control you have is where (in which cell) do you want to run a process or key component

Memory Locality

- On large Integrity servers with multiple quad building blocks it is important to be close to where the ”action” is.
- The dedicated lock manager is the top user of the LCKMGR spinlock, so it should be running on a CPU in the same quad as the memory for the spinlock itself.
- TCPIP CPU should be where the global TCPIP spinlock is.
- If you have applications which make heavy usage of ASTs then affinitize them to CPUs in the quad where the SCHED spinlock memory lives

Memory vs Disk

- Todays large systems tend to have lots of physical memory
  - Use it !
- Make things larger
  - Quotas and limits
  - Some programmer still worry about bytes, make them quadwords and you wont have to worry about alignment faults
- Install heavily used images and shareable images resident
  - Install with shared address space
  - Paged pool requirements
- Avoid faulting in pages from disk
  - Memory access are orders of magnitude faster than doing a disk I/O

Alignment Faults

- ***Eliminate alignment faults !!.***
- They are bad on Alpha and they are very bad on Itanium !
- On Itanium the OS has to fix them up, needs MMG spinlock
- MONITOR ALIGN will show the rates
  - 10,000 alignment faults per second is potentially a problem
  - 100,000 alignment faults per second is a problem
  - Fixing them will result in noticeable performance improvements
- The routine detecting the alignment fault is not always the guilty party
  - Many times un-aligned data is passed into routines

MAKLEE

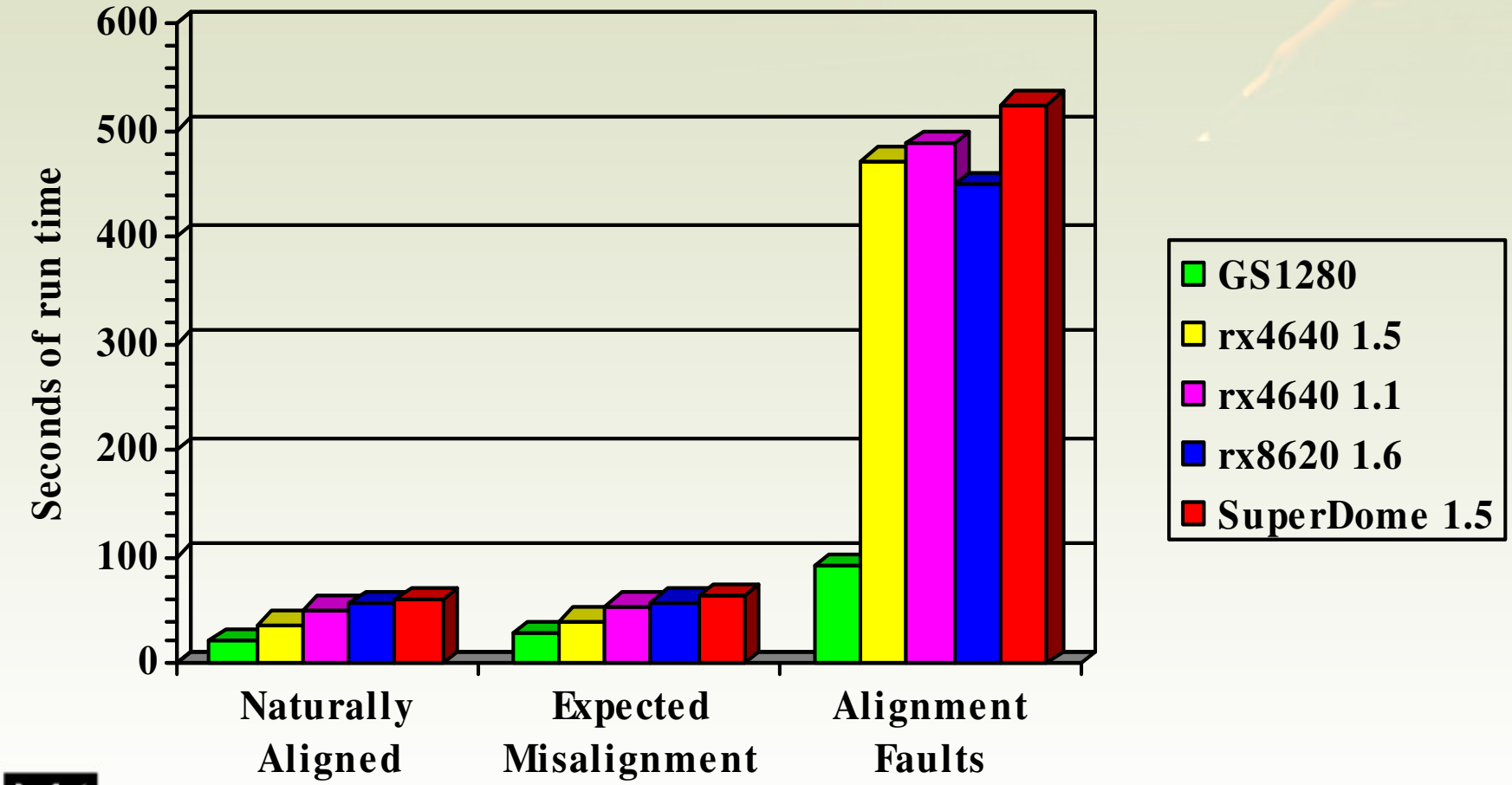
Alignment Faults

```
$ monitor align
```

```
                    ALIGNMENT FAULT STATISTICS
                          on node XYZW1
                      14-SEP-2009 19:07:24.23

                                  CUR        AVE        MIN        MAX

Kernel Fault Rate                0.00       0.33       0.00       0.99
Exec   Fault Rate               19.98     164.83      19.98     283.71
Super  Fault Rate                1.99     552.78       1.99    1654.36
User   Fault Rate           332452.31  332447.22  331692.00  333197.40

Total  Fault Rate           332474.28  333165.51  332474.28  333537.18
```

MAKLEE

Alignment Faults (cont'd)



MAKLEE

DECPS

- Consider stopping DECPS.
- Very intrusive product.
- Slows down every I/O on the system.
- Stopping DECPS will provide ~ 10% performance increase.

MAKLEE

Dedicated Lock Manager

- On large systems with heavy locking activity, it is more efficient to dedicate a single CPU to perform all locking operations
  - Less contention for LCKMGR spinlock
  - Less dirty cache reads
  - Does local $ENQ and $DEQ operations
  - Distributed lock manager in a cluster still uses LCKMGR spinlock
  - $GETLKI does not call the dedicated lock manager
- Recent fix to V8.3-1H1 has some performance improvements for the dedicated lock manager
  - It prefetches cache lines of the lock request packets to avoid stalling

”The Limits”

- Avoid ”Hitting” the limits
  - Primary CPU 100% busy
    - Certain things still only run on the primary CPU, like timers
    - Solution: offload anything which can run on other CPUs
  - Dedicated lock manager 100% busy
    - Reduce locking (might not be possible)
  - TCPIP CPU is 100% busy
    - Enable Jumbo frames
    - Enable PPE in TCPIP V5.7
    - Enable local I/O post-processing

MAKLEE

”The Limits”

- Spinlock hold time close to 100%
  - Find out which spinlock and what routine (SPL tracing)
  - If MMG and physical data reads then reduce alignment faults
  - If SCHED and high AST rate then find a way to reduce the AST activity (might need application changes)

HyperThreading

- Enabling HyperThreading on systems with:
  - High CPU utilization
  - Compute queue
  - Applications with poor locality

- HyperThreads will allow process A to use the CPU (core) while process B performs a memory fetch.

Questions?

See us at www.maklee.com for:

- Performance tuning
- Oracle Tuning
- Platform Migration
- Custom Engineering
- Custom Training

…..Bis bald

MAKLEE

ORACLE PARTNER