(Advanced) OpenVMS Performance Tips & Tricks

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

Basic Tuning Techniques

- OpenVMS V8.3-1H1
  - VMS831H1I_SYS-V0700
- SET RMS
- QUANTUM
- Resident Images
- Cache vs. No cache
- Fastpath
- Compile /optimize
- Hyper Threads
- PE data compression
- Gigabit Jumbo Frames
- SDA PRF

MAKLEE

Quote

- “Keep looking below surface appearances. Don't shrink from doing so just because you might not like what you find.” Colin Powell

MAKLEE

VHPT

- Virtual Address lookup IA64
  - CPU TB cache
  - VHPT
  - OpenVMS performs 3 level address translation walking the page tables.
- The VHPT is sized by SYSGEN parameter - VHPT_SIZE.
- Default value of 1 means allocate 32KB per CPU for the VHPT.

Oracle Batch job A

MAKLEE

Elapsed Time in Minutes (less is better)

VHPT

- No good deed goes unpunished.
- High cost associated with invalidating large address space.
  - Oracle server process mapping large SGA
- May result in high MP Synch time during while invalidation is in progress.
- Processes may show up in RWSWP.
- Large VHPT not suitable for applications that frequently map large virtual address space for short period of time.
- In severe situations stop all CPUs on the system until condition clears up.
  - Shutting down Oracle database requires 25 minutes vs. 3.

Multiple Kernel Threads

- When running a threaded application, the threads manager creates one Kernel thread per CPU.
  - This happens regardless of the number of user threads in the application
- Kernel threads are execution engines for user threads
- The threads manager schedules user threads to run on an available kernel thread.
- Overhead is associated with managing multiple Kernel threads.
- A threaded application decides if multiple kernel threads should be enabled or disabled.
- With kernel threads disabled – one execution engine is used for running the user threads.

Multiple Kernel Threads

- Evaluated the impact of disabling multiple kernel threads on a Java based benchmark.
- Single threaded Java program performing CPU intensive operation (encryption).
- SD32B, 32 CPUs, OpenVMS V8.3-1H1, Java 5.
- Used SET IMAGE to disable multiple kernel threads.

Encryption Test

- Multiple kernel threads (MKT) enabled

```
Accounting information:
Buffered I/O count:              103709      Peak working set size:     891216
Direct I/O count:                  7279      Peak virtual size:        2652928
Page faults:                      55739      Mounted volumes:                0

Charged CPU time:        0 00:02:36.81      Elapsed time:       0 00:15:54.98
```

- Multiple kernel threads (MKT) disabled

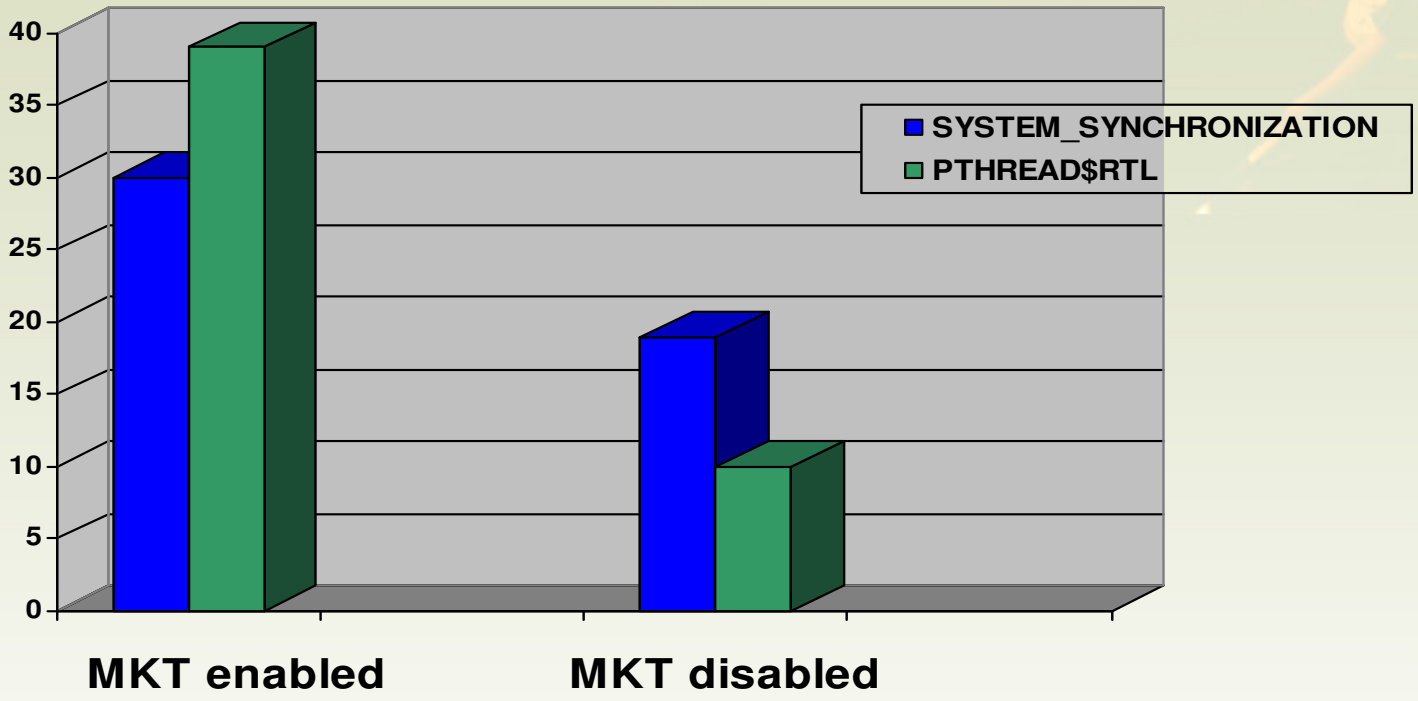
```
Accounting information:
Buffered I/O count:              102399      Peak working set size:     841424
Direct I/O count:                  7145      Peak virtual size:        2584064
Page faults:                      52623      Mounted volumes:                0

Charged CPU time:        0 00:01:35.80      Elapsed time:       0 00:15:18.83
```

  - 39% less CPU time

MAKLEE

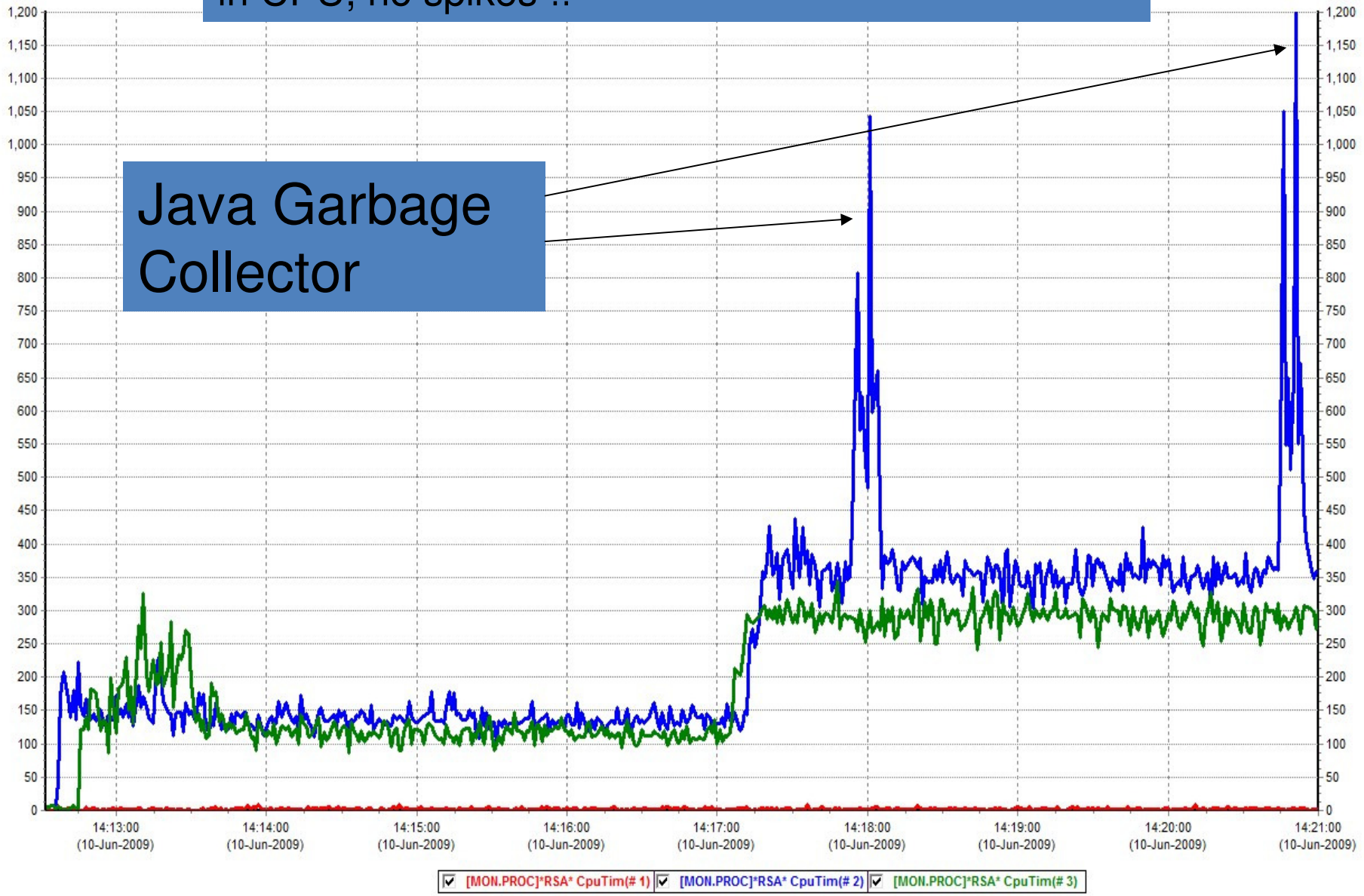
PC Sampling



Number of CPU cycles (in percent of total)
Less is better

MAKLEE

CPU utilization during benchmarks.15% reduction in CPU, no spikes !!



Data Encryption

- Business rules and data privacy regulations force more and more organizations to encrypt data stored on tapes.
- Starting with OpenVMS V8.3, OpenVMS can generate encrypted savesets.
  - OpenVMS supports various AES encryption algorithms, and various encryption key sizes.
- OpenVMS also supports the LTO-4 tape drive family.
  - LTO-4 tape drives support hardware encryption.

Which one would perform better?

Encryption Benchmark

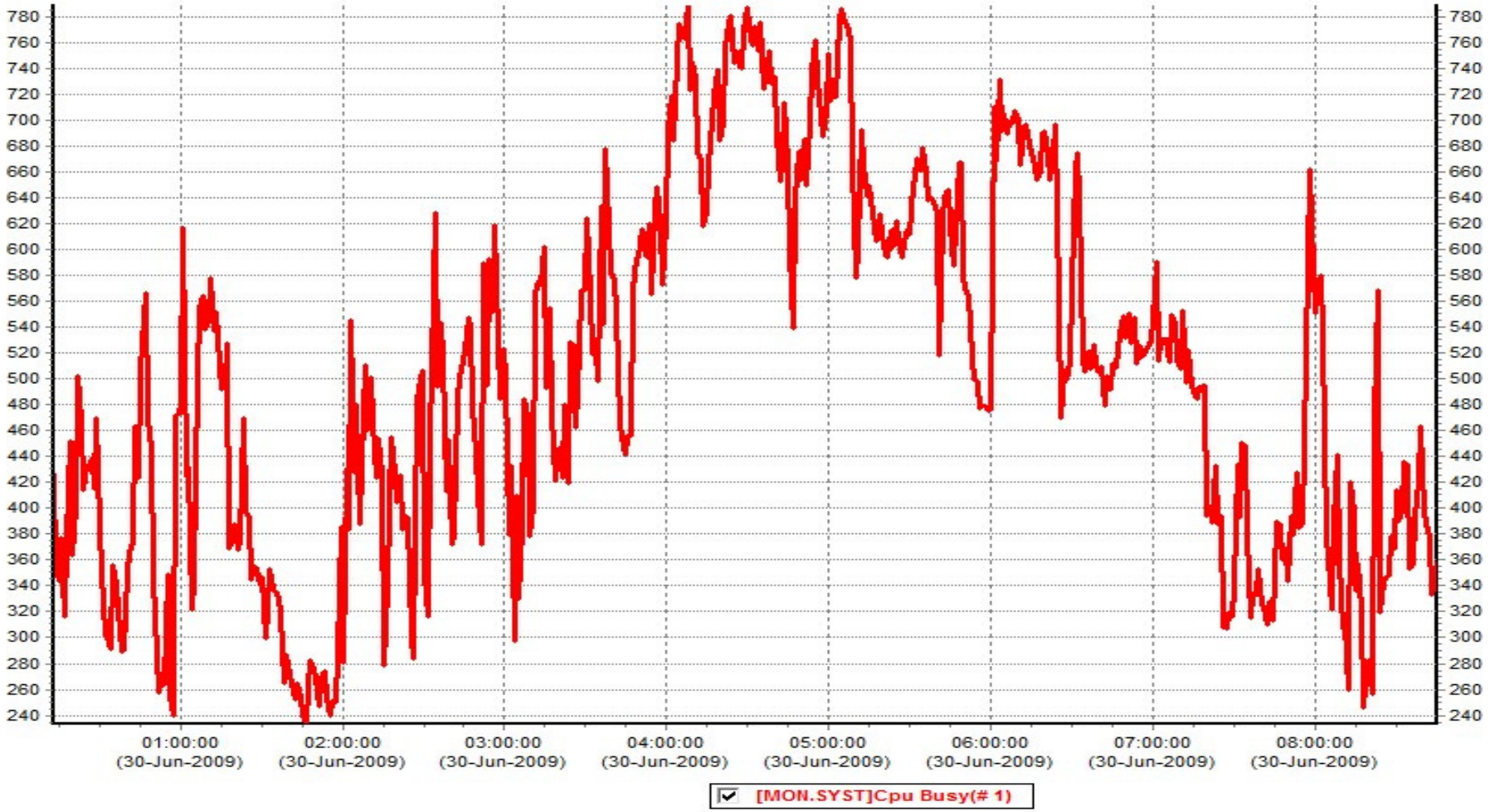
- Customer benchmark comparing performance of:
  - Alphaserver ES80, 8 CPUs
  - EVA 8100
  - OpenVMS V8.3
  - 2gb fiber connection
  - LTO-3

  VS.

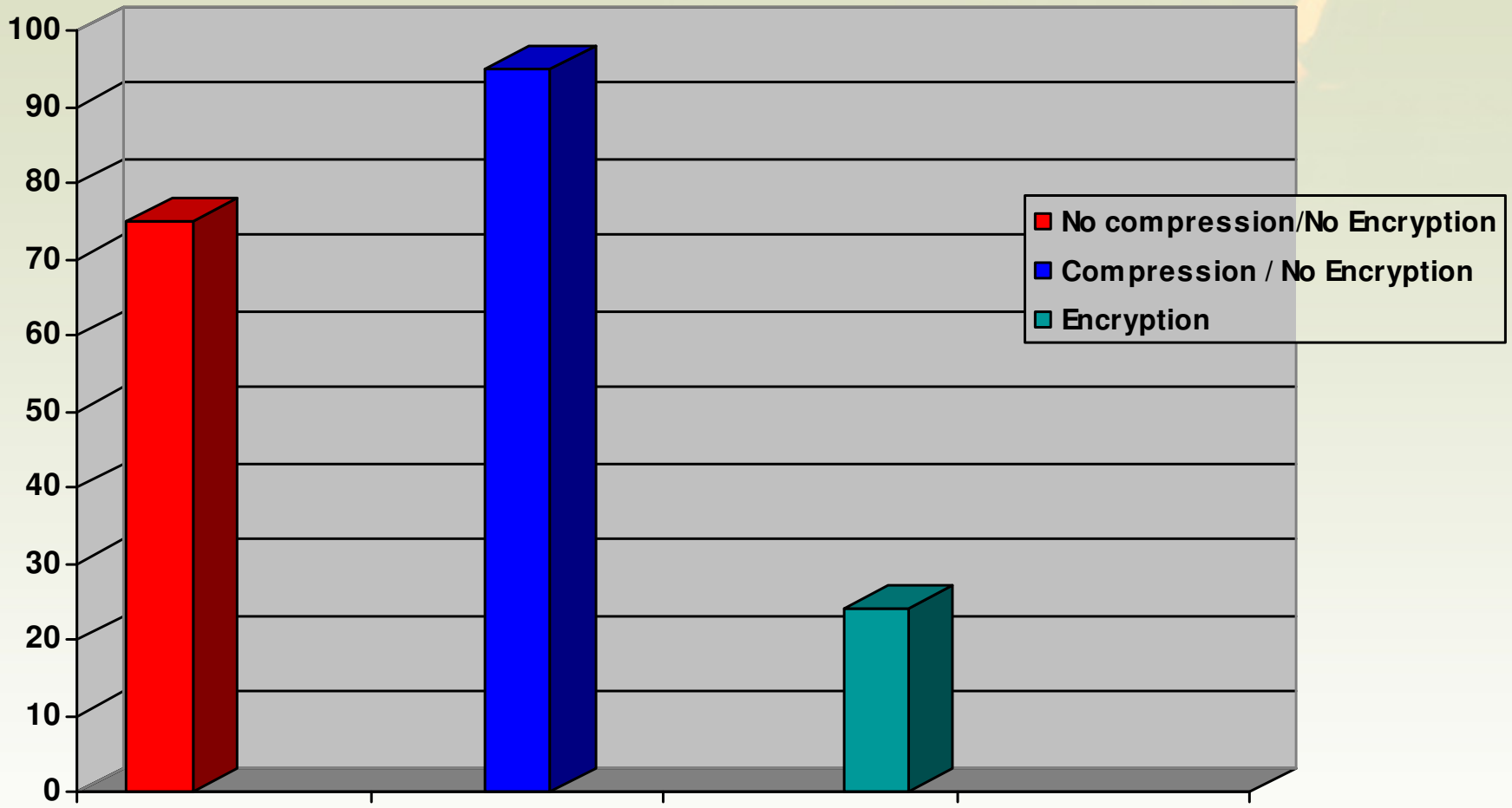
  - 4P/8C BL870c
  - EVA 8100
  - OpenVMS V8.3-1H1
  - 4 gb fiber connection
  - LTO-4

MAKLEE

CPU utilization during backup



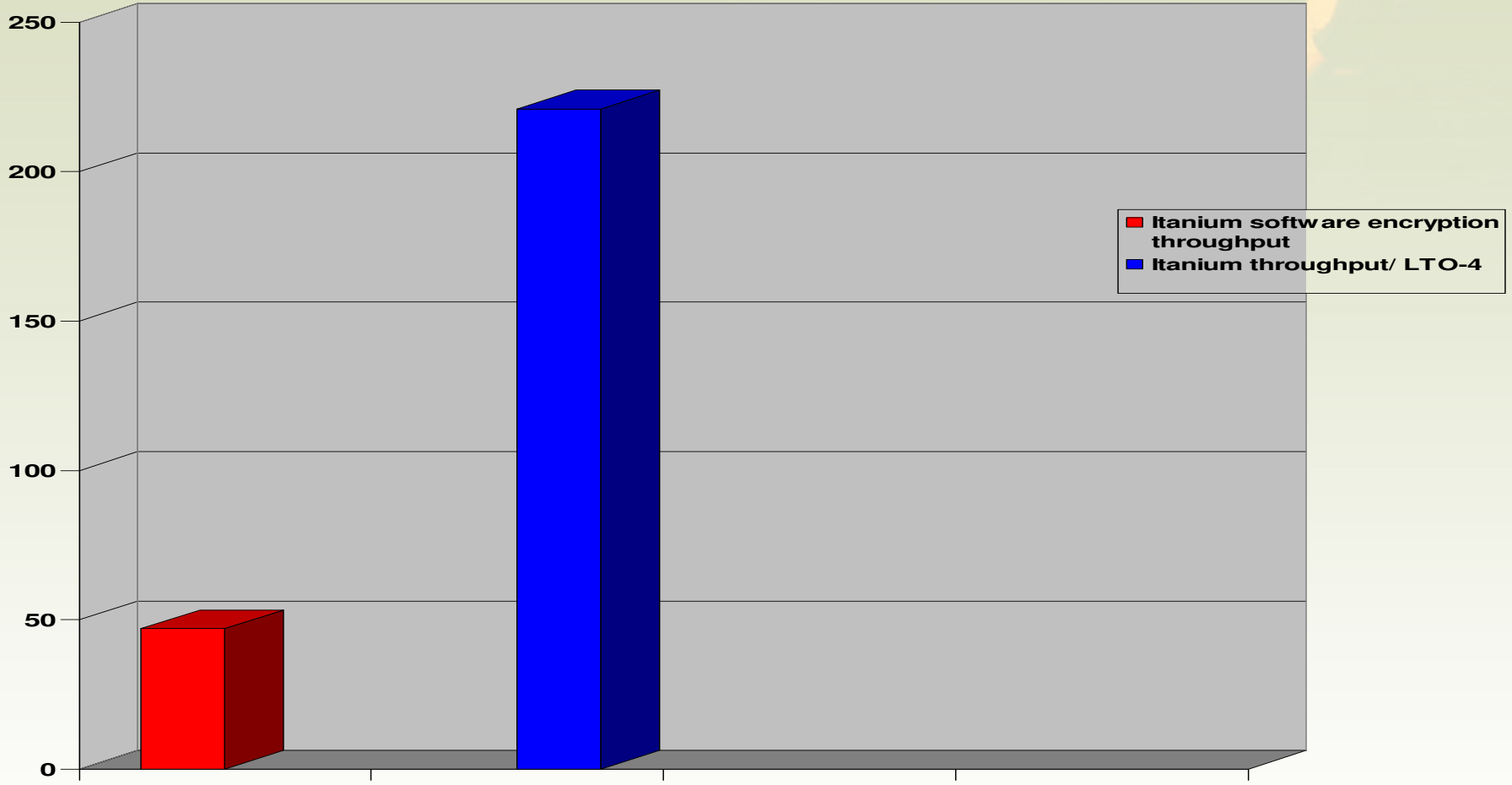
AlphaServer ES80 Throughput



MAKLEE

LTO-3 throughput MB/sec
More is better

BL870 Throughput



MAKLEE

LTO-4 throughput MB/sec
More is better

Data Encryption

- Always enable hardware data compression.
- Hardware encryption outperforms software encryption.
- Stronger encryption keys require more CPU resources.
- Depending on the storage sub system, the /IO_LOAD qualifier may improve performance of backup operations.
- The MSL tape library allows distributing backup across more physical tapes, increasing the throughput of the backup operation.

MAKLEE

Mount

- More and more systems use large number of disk volumes.
  - Overcome VMS limitation of 1TB per volume.
- It is not unheard of to encounter systems with 100+ volumes.
- Mounting volumes is slow !!
- Try parallelising the mount operation.
  - SPAWN is the easiest way
  - Writing a program calling $MOUNT is the fastest way

Mount

- BL870c.
- OpenVMS V8.3-1H1.
- DT cluster spread over 2 sites, 5KM apart.
- Booting the system required 15 minutes.
  - Sequantially mounting 100 shadow sets.
- After changing the startup script to mount all volumes in parallel, the system now boots in 1.5 minutes.

MAKLEE

SYS$IO_PERFORM

- One of VMS's best kept secrets.
- SYS$IO_PERFORM starts a fast I/O operation.
  - Developed as an alternative to SYS$QIO.
- Shortcut into the core of the I/O subsystem.

MAKLEE

Fast Copy

- Fast I/O is the fastest way to copy data from one disk to another.

- The following shows the results of various tests copying 500MB file from one disk to another:

CONFIG	METHOD	~DIO	CPU	ELAPSED
EVA3000	COPY	16156	02.71	00:12.94
LP9802	BACKUP	31288	00.72	00:14.06
	BACK/BLO=65024	15911	00.35	00:12.49
GS1280	CONVERT	23529	08.97	00:15.37
VMS V8.2	**FAST_IO_COPY**	**7845**	**00.22**	**00:07.34**

Contact me off-line for a copy of the fast copy program.

MAKLEE

C Vs. C++

- The C compiler uses a backend (code generator) provided by HP.
- The C++ compiler uses a backend provided by Intel
  - The C++ compiler knows how to use the Itanium advanced loads and speculative loads.
  - Allows the compiler to hoist fetches out of loops AND move fetches before stores that might impact them
  - Can be huge performance win for certain applications.
- Try compiling CPU intensive C routines with the C++ compiler.
  - We've seen ranges from the C compiler is 20% faster to Intel's compiler is twice as fast.

YMMV !!

TCP/IP I/O post processing

- TCP/IP interrupts are handled by one CPU.
  - SHOW FASTPATH will display the TCP/IP CPU.
- Saturating the TCP/IP CPU will limit the throughput of the application.
- Busy systems with heavy TCP/IP traffic should enable local I/O post-processing for TCP/IP.
  - I/O post processing will be performed on the CPU issued the I/O vs. the TCP/IP CPU.
  - Off loads the TCP/IP CPU.
- To enable local I/O post-processing for TCPIP
  - sysconfig –r net ovms_unit_status = 2147483648
  - Add to sysconfig.tab
- Watch out for the PPE feature in TCP/IP V5.7

TCP/IP FTP

- Use the following logical names to speed up FTP transfers:
  - TCPIP$FTP_FILE_ALQ
  - TCPIP$FTP_FILE_DEQ
  - TCPIP$FTP_WNDSIZ
- Logical names may be set system wide or only for specific processes.

MAKLEE

Oracle RAC

- Oracle RAC startup/shutdown is very slow.
  - Rx7640, OpenVMS V8.3-1H1, Oracle 10gR2 requires 29 minutes to start 10 RAC databases under CRS.
  - CRS startup and shutdown is serialized.
  - Try parallelizing the startup using PIPE
    - Disable automatic startup of all databases
    - PIPE start db1 | start db2 | start db3 …
  - The system in question now starts all 10 databases in 4.5 minutes.

Data Pump

- Data pump is the fastest way to export/import data.
- Data pump creates multiple threads lowering elapsed time required for export/import to complete.
- Data pump on OpenVMS will gradually slow down as the dump file grows.
- Use the parallel=n feature to guarantee single dump file does not grow beyond 1GB.

MAKLEE

PRF

```
SDA> prf load
PRF$DEBUG load status = 00000001
SDA> prf start pc/ind=21E004DA
PC Sampling started...
SDA> prf start collect
SDA>
Now run the application:

$ r prime
 ELAPSED:     0 00:00:24.16  CPU: 0:00:24.06  BUFIO: 0  DIRIO: 0  FAULTS: 0
$
```

- To look at the collected data:

```
SDA> prf show collect
```

MAKLEE

PRF SHOW COLLECT

Start VA	End VA	Image	Count	Percent
FFFFF802.11F00000	FFFFF802.11F01FFF	PRIME	305113	99.85%
FFFFF802.A1000000	FFFFF802.A1015FFF	Kernel Promote VA	1	0.00%
FFFFFFFF.80000000	FFFFFFFF.800000FF	SYS$PUBLIC_VECTORS	2	0.00%
FFFFFFFF.80000100	FFFFFFFF.800111FF	SYS$BASE_IMAGE	2	0.00%
FFFFFFFF.80011200	FFFFFFFF.800651FF	SYS$PLATFORM_SUPPORT	258	0.08%
FFFFFFFF.800A0000	FFFFFFFF.801DD6FF	SYSTEM_PRIMITIVES	88	0.03%
FFFFFFFF.801DD700	FFFFFFFF.80243BFF	SYSTEM_SYNCHRONIZATION_MIN	9	0.00%
FFFFFFFF.80254600	FFFFFFFF.8026EFFF	SYS$EIDRIVER.EXE	5	0.00%
FFFFFFFF.8026F000	FFFFFFFF.802895FF	SYS$LAN.EXE	2	0.00%
FFFFFFFF.80289600	FFFFFFFF.802BA1FF	SYS$LAN_CSMACD.EXE	2	0.00%
FFFFFFFF.80440E00	FFFFFFFF.8052B2FF	IO_ROUTINES	1	0.00%
FFFFFFFF.8053A600	FFFFFFFF.80670DFF	PROCESS_MANAGEMENT	7	0.00%
FFFFFFFF.80670E00	FFFFFFFF.807759FF	SYS$VM	11	0.00%
FFFFFFFF.80779500	FFFFFFFF.807C76FF	LOCKING	1	0.00%
FFFFFFFF.807C7700	FFFFFFFF.807F9CFF	MESSAGE_ROUTINES	1	0.00%

MAKLEE

PRF SHOW COLLECT

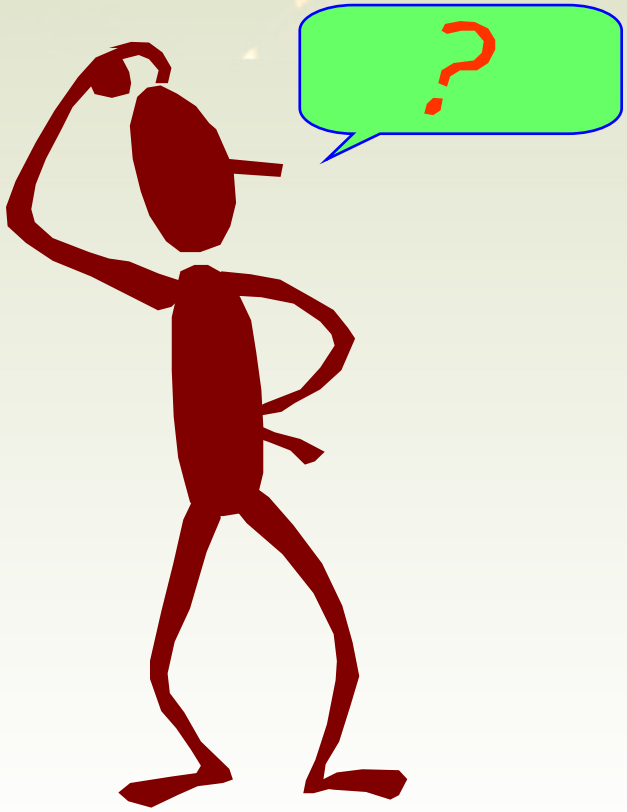
```
SDA> prf show coll/threash=2
PC                  Count       Rate   Symbolization                              Module                     Offset
-----------------  -------  ---------  ----------------------------------------  -------------------------- --------
FFFFF802.11F00170      63410     20.07%  PRIME+10170                               PRIME                      00010170
                                         [GENERATE_PRIME+00000170 / GENERATE_PRIME+00000170]
FFFFF802.11F00190       6138      2.01%  PRIME+10190                               PRIME                      00010190
                                         [GENERATE_PRIME+00000190 / GENERATE_PRIME+00000190]
FFFFF802.11F001A0       6761      2.21%  PRIME+101A0                               PRIME                      000101A0
                                         [GENERATE_PRIME+000001A0 / GENERATE_PRIME+000001A0]
FFFFF802.11F00200       6296      2.06%  PRIME+10200                               PRIME                      00010200
                                         [GENERATE_PRIME+00000200 / GENERATE_PRIME+00000200]
FFFFF802.11F00220       8102      2.65%  PRIME+10220                               PRIME                      00010220
                                         [GENERATE_PRIME+00000220 / GENERATE_PRIME+00000220]
FFFFF802.11F00290       6804      2.23%  PRIME+10290                               PRIME                      00010290
```

MAKLEE

Questions? Comments?

Would never have been possible without the gracious & expert assistance of:

Christian Moser & Norman Lastovica



MAKLEE