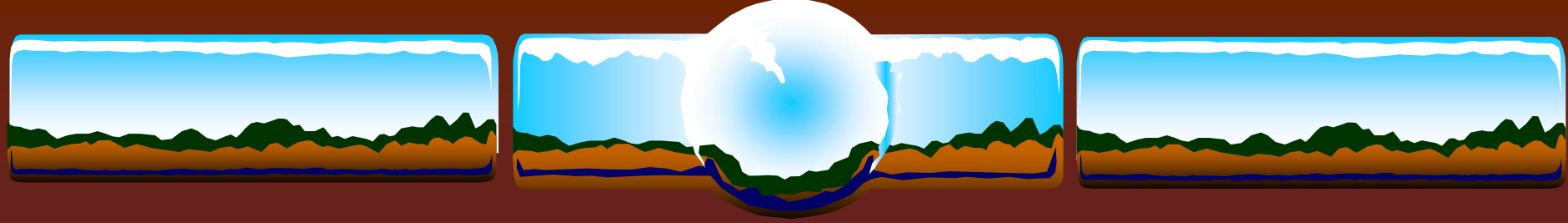
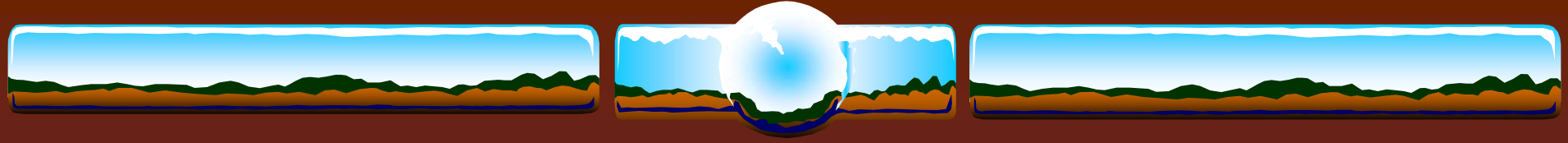


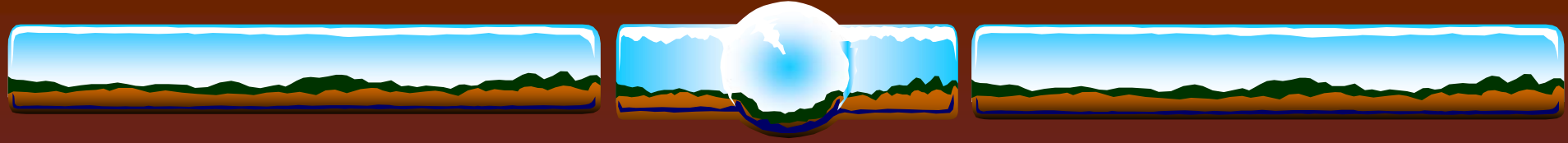
OpenVMS Information Desk

Guy Peleg / Norman Lastovica

2 November 2004



The Secrets of Performance

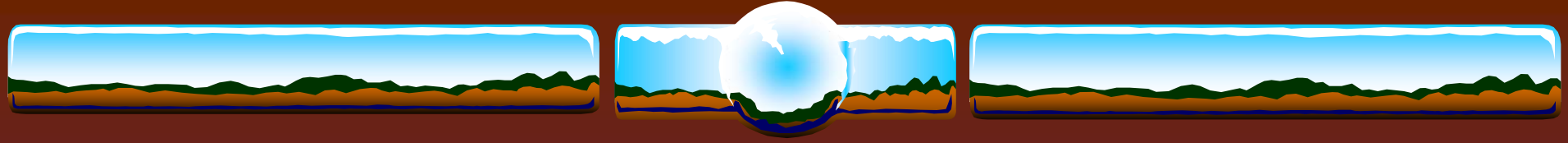


Our Golden Rules

The best performing code is
the code not being executed

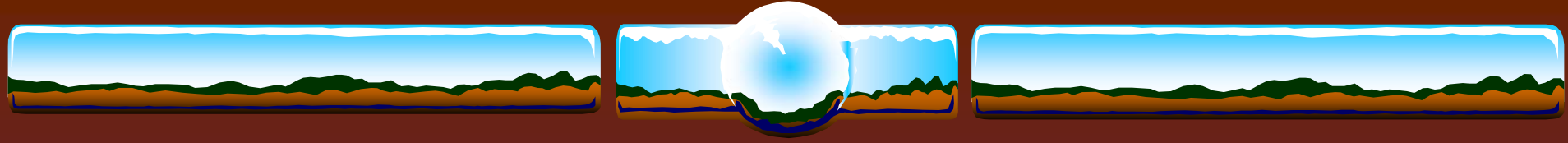
The fastest I/Os are those avoided

Idle CPUs are the fastest CPUs



VMS Versions

- V7.3-1
  - “Required” for > 4 CPUs
  - Dedicated lock manager, scheduling improvements, fastpath SCSI and FIBER
- V7.3-2
  - Better & faster
    - Working set in S2, per mailbox spinlocks, per PCB spinlocks, LAN fastpath, scalable TCPIP kernel



.....and most important.....

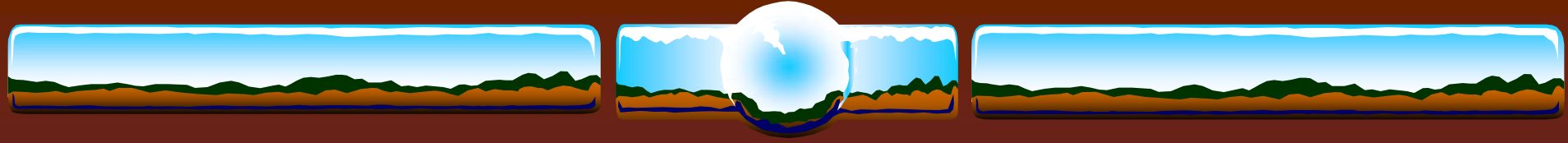
Many new DCL features

Configuration

- Dedicated CPU Lock Manager
  - Keep it dedicated!
- FastPath
- Path balance
- I/O Adaptors / QBB
- Write-back cache
  - On controllers where available
  - On devices where practical
    - Manually/explicitly set flags in most disks; Usually only viable for locally connected SCSI disks

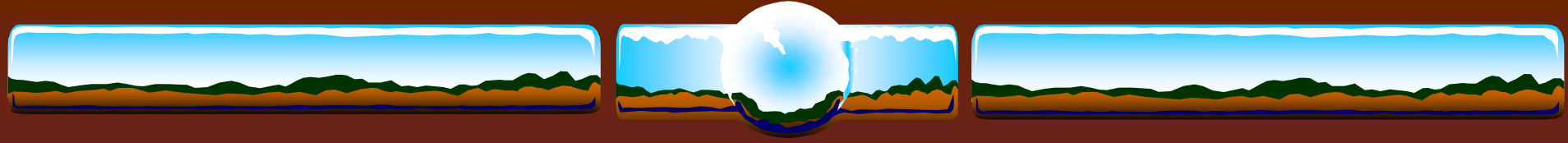
Locking

- Remember that remote lock operation can be slower than local lock operation
- Balance LOCKDIRW based on CPU power
  - GS1280s clustered with a VAX 6440
- MIN_CLUSTER_CREDITS=128 for big/fast machines
- DEADLOCK_WAIT=1
  - This isn’t 1982 any longer



Wildfire

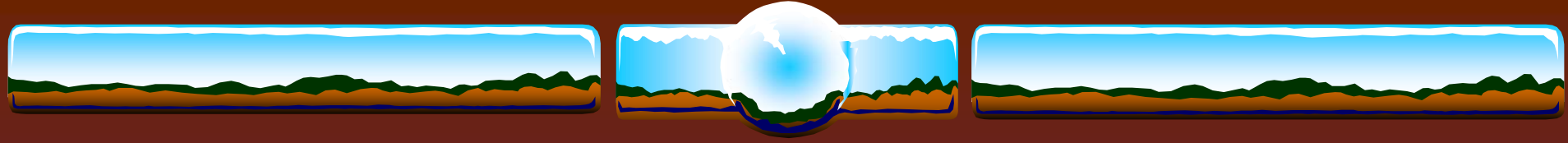
- Keep memory close to processors as much as possible
  - Install images /RESIDENT if they are used by many processes or are performance critical
  - **`SDA> SHOW EXEC /SUMMARY`** and make sure executive images are “sliced”
  - Evaluate RAD-specific processes/global sections
  - Memory reservation
    - XFC, Pool



Marvel

- Better, Faster, Stronger
- RADs are likely not a worry
- “Don’t sweat the NUMA”

  - *Steve Hoffman Oct. 15th 13:29*

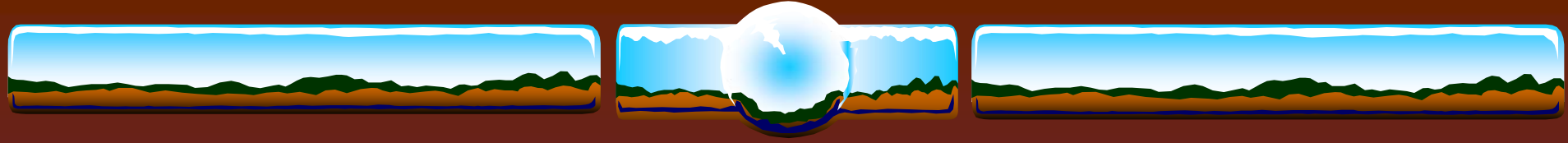


Transition Slide

"If you change nothing you can be sure that performance won't improve" - *Norm Lastovica Oct. 15th 12:01*

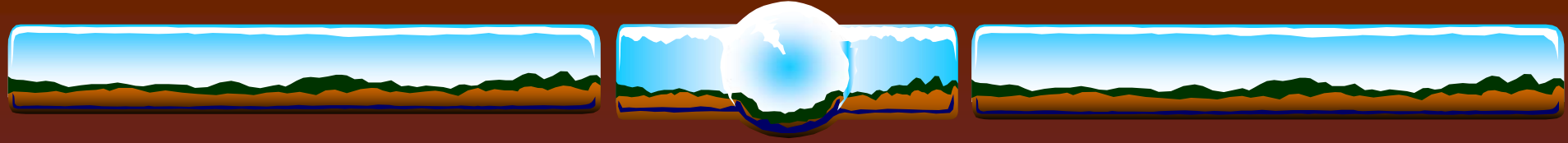
"Buying newer hardware is the least risky way of improving performance" - *Norm Lastovica Oct. 15th 12:03*

"Application changes have the greatest potential of improving performance" - *Guy Peleg Oct. 15th 12:05*



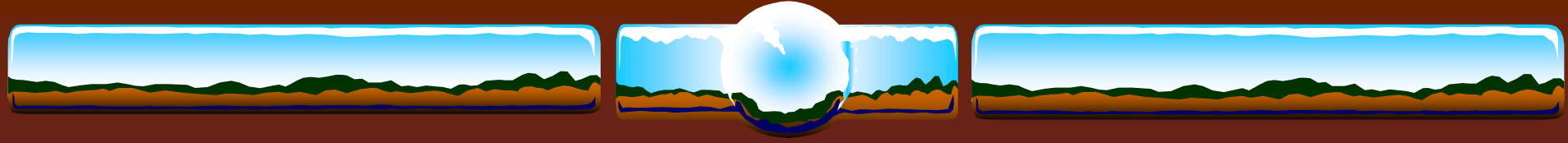
/NOOPTIMIZE

- Typically for debugging
- Many more memory references for local variables
- Longer instruction stream - “One thing at a time”
- Sometimes used to work around program bugs



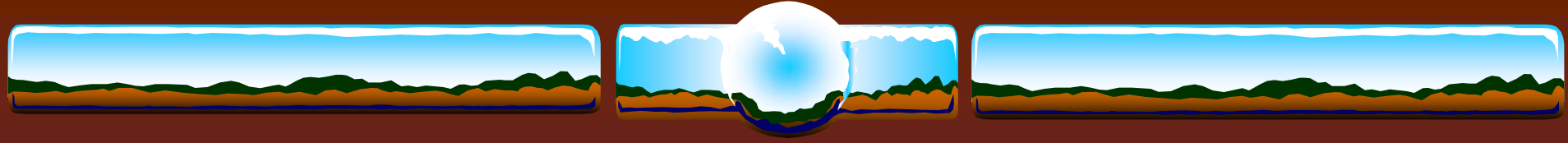
/OPTIMIZE

- Instructions “spread” though many source lines
- Avoids memory references for local variables
- Faster instruction sequences - “Multiple things at once”
- “Unrolled” loops to avoid branches
- Several options (based on language)
  - Optimization “level”, Alignment assumptions, Atomicity assumptions, “UNROLL” counts, Routine “inlining”, Aggressive pipelining



/OPTIMIZE=…TUNE=

- Code sequences *biased* towards scheduling characteristics of specified processor; Runs on all generations
- Can produce code to make run-time decisions
  - AMASK / IMPLVER to detect processor capabilities
  - Generate multiple code sequences
  - Use “better” sequences where worthwhile based on CPU



/ARCHITECTURE=

- Generate code for specified architecture *and later*
- Optimal instruction scheduling
- Use of all available instructions

Examples of …TUNE & /ARCHITECURE

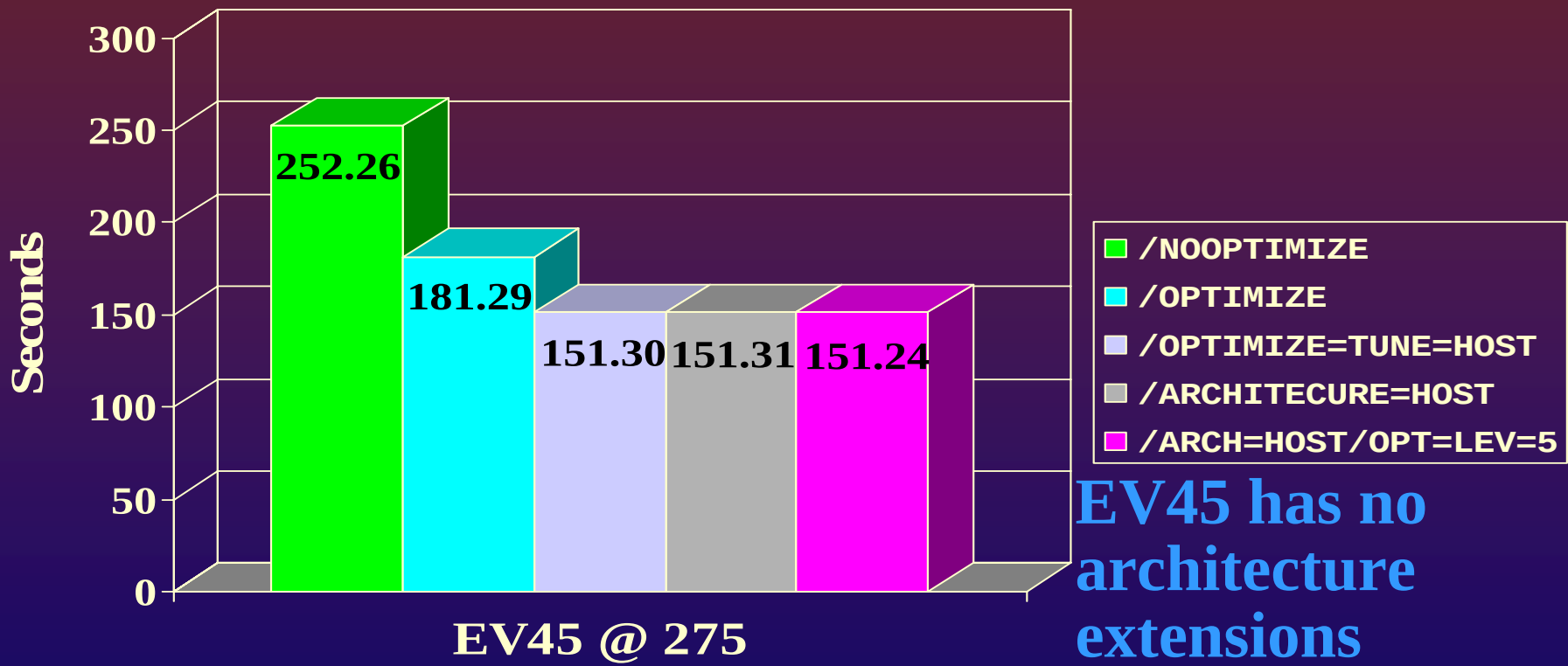
- **/OPTIMIZE=TUNE=EV56**
  - Execute on all Alpha generations
  - Biased towards EV56
- **/OPTIMIZE=TUNE=EV6 /ARCHITECTURE=EV56**
  - Execute on EV56 and later (Byte/Word instructions)
  - Biased for EV6 (quad issue)
- **/ARCHITECTURE=EV6**
  - Execute on EV6 and later (Integer-Floating conversion, Byte/Word & Quad-issue scheduling)
- **/ARCHITECTURE=HOST**
  - Code intended to run on processors the same type as host computer
  - Eexecute on that processor type and higher

Prime Numbers Test

- Find first 1,000,000 prime numbers

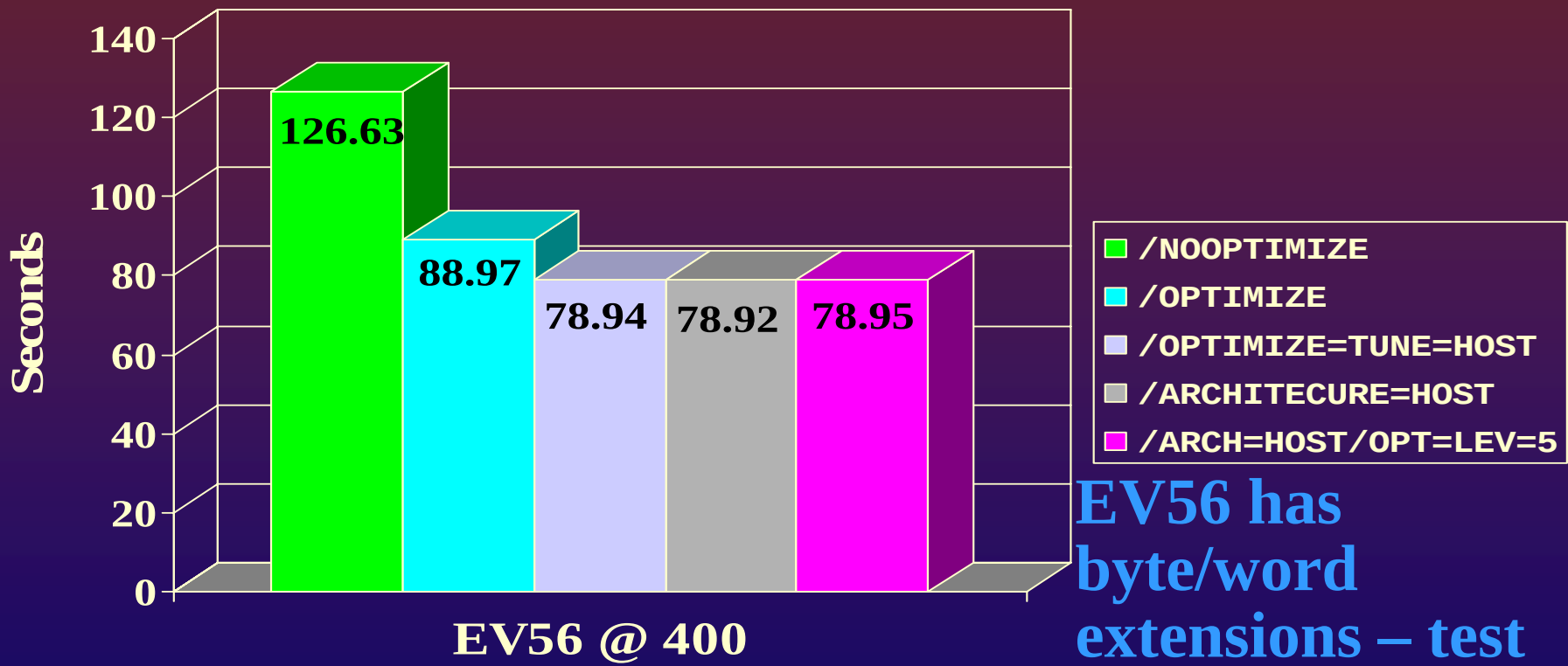
```
        primes(1) = 3
        hi_prime = 3
        hi_prime_index = 1
        hi_prime_divisor_index = 1
        do 100 i = 5,2000000000,2
        if (primes(hi_prime_divisor_index)**2 .lt. i)
            hi_prime_divisor_index = hi_prime_divisor_index + 1
        do 20 j = 1, hi_prime_divisor_index
        if (mod(i, primes(j)) .eq. 0) go to 100
20      continue
        hi_prime_index = hi_prime_index + 1
        primes(hi_prime_index) = i
        hi_prime = i
        if (hi_prime_index .eq. n_primes) go to 200
100     continue
200     …
```

Generating Primes
AlphaServer 2100 4/275



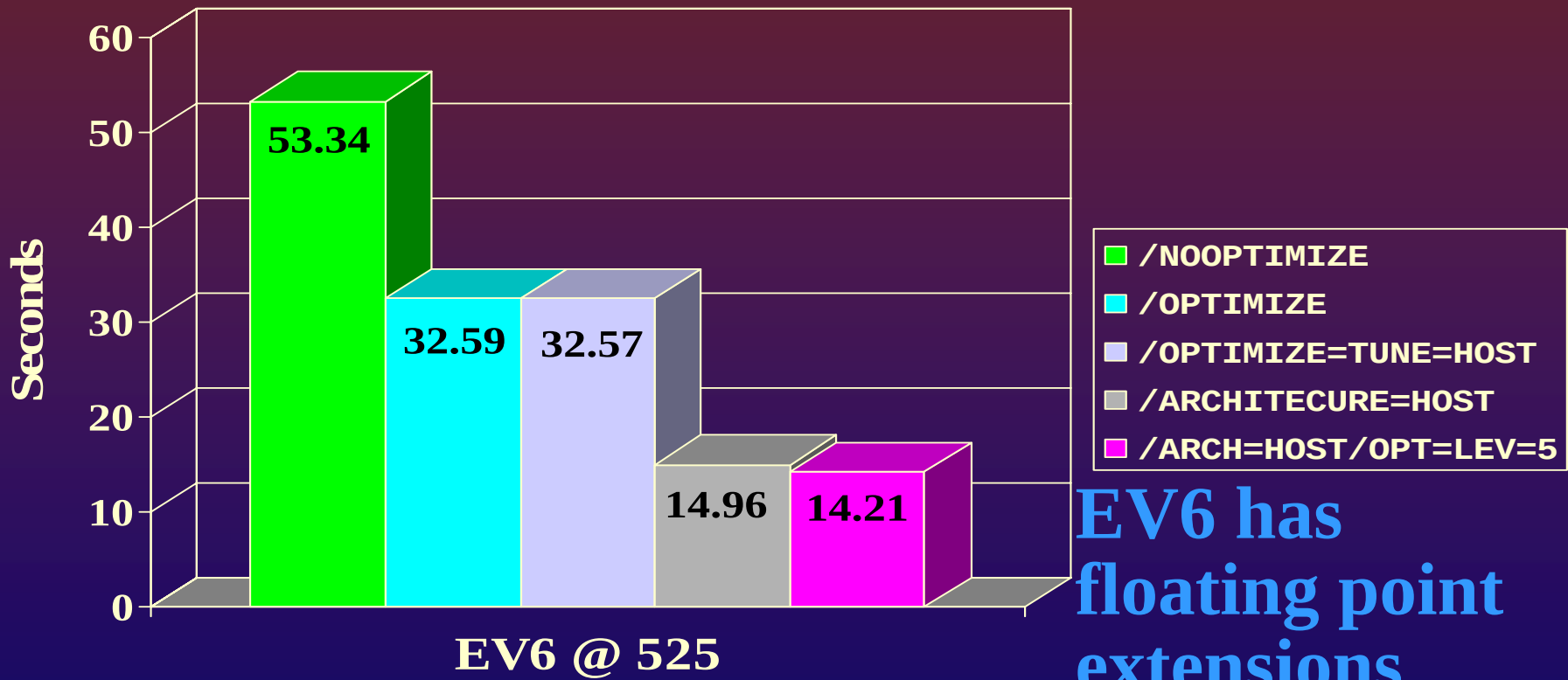
EV45 has no architecture extensions

Generating Primes
AlphaServer 4100 5/400



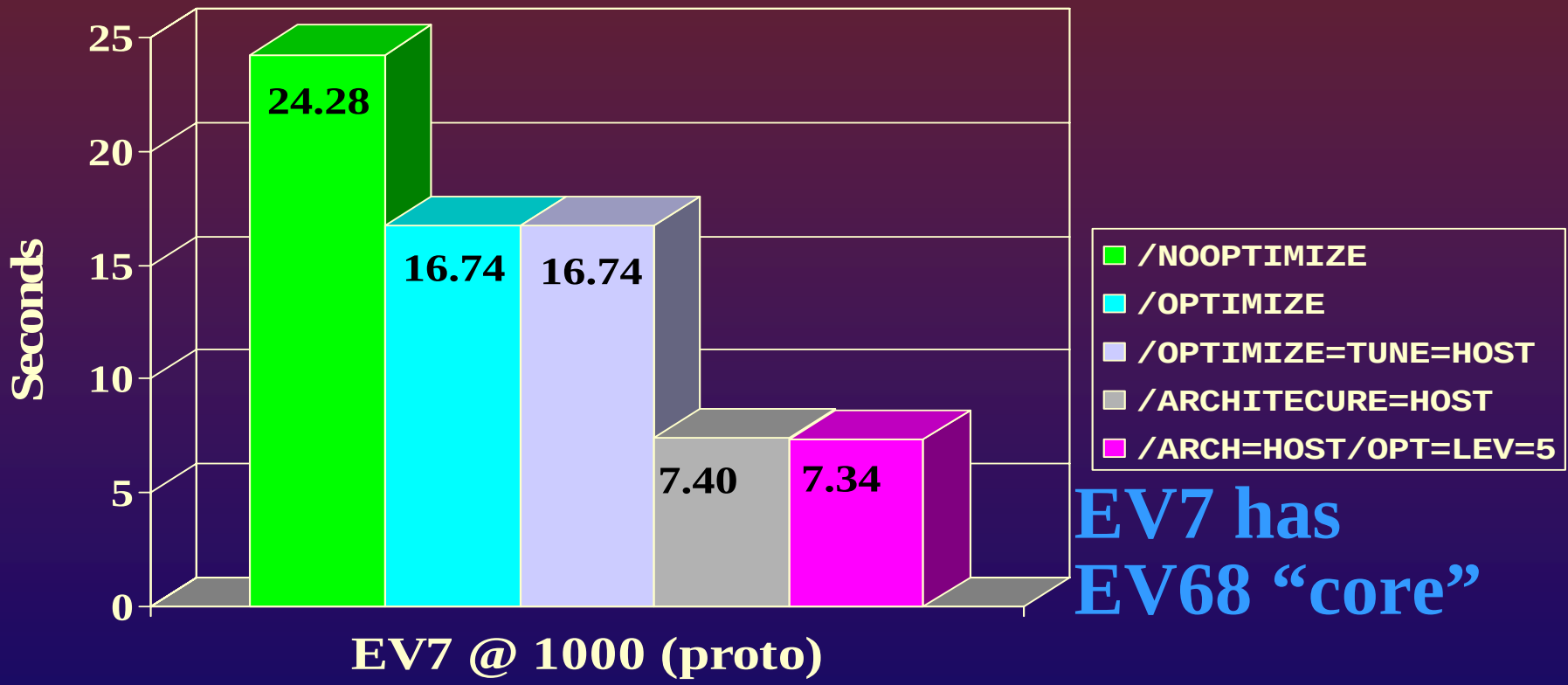
EV56 has byte/word extensions – test does not use them

Generating Primes
AlphaServer GS140 6/525



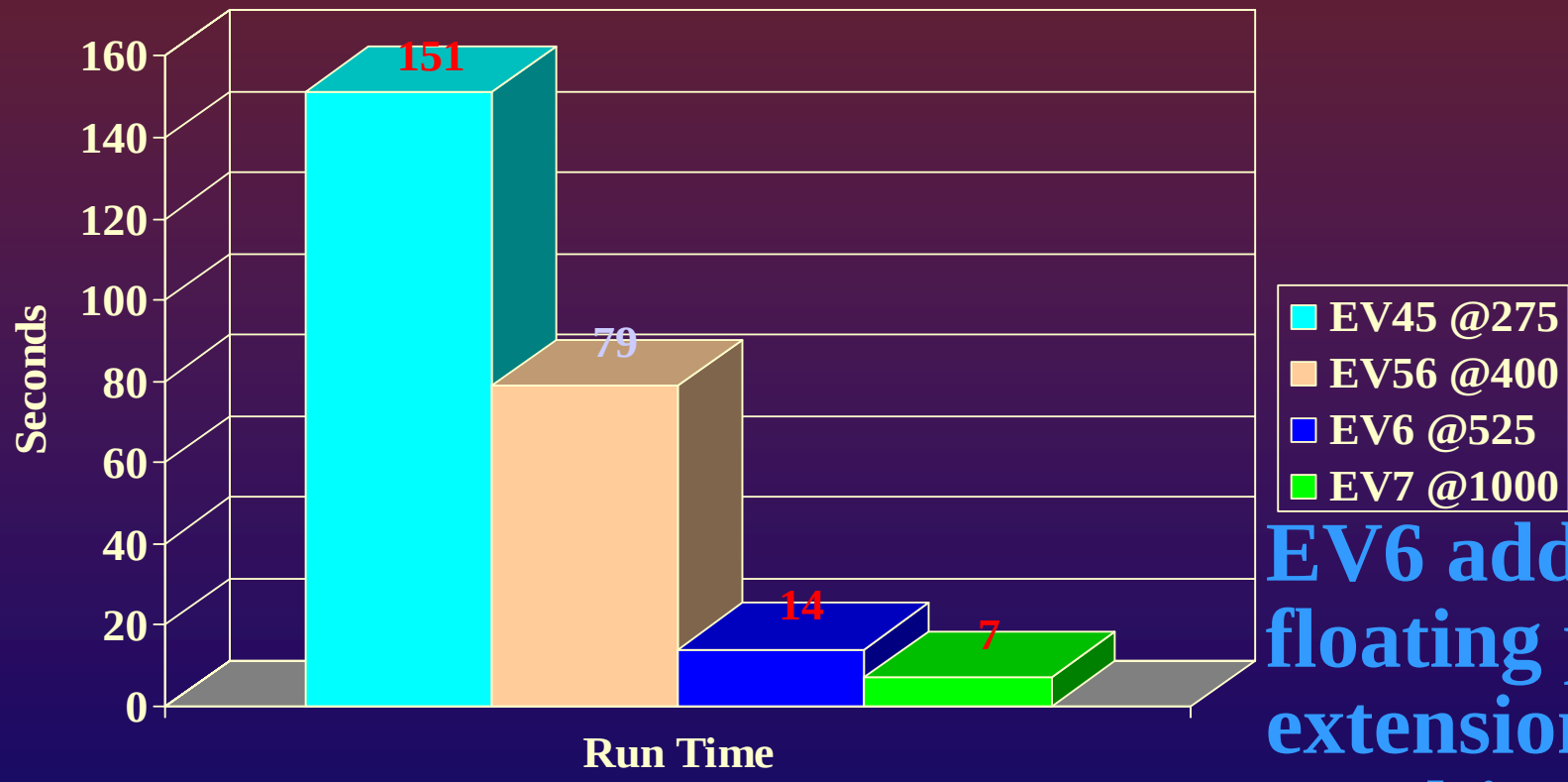
EV6 has floating point extensions

Generating Primes
GS1280 7/1000 (prototype)



EV7 has EV68 “core”

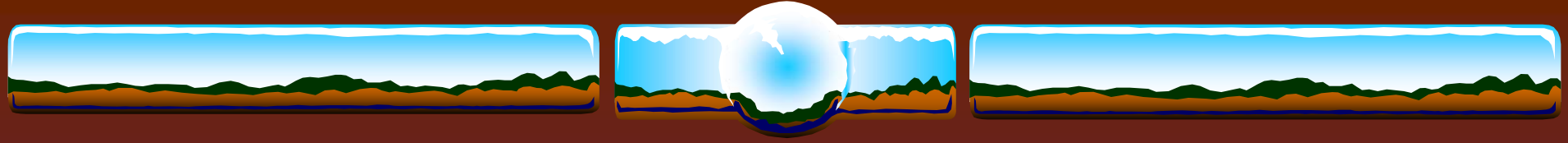
Generating Primes… Comparing the Machines



EV6 adds floating point extensions & quad-issue

Real-life Example /OPTIMIZE

- Commercial Trading system
  - Inserts ~2 rows per trade into database
- >99% CPU bound
- 90+% user mode time
  - Performing extensive trade validations
  - < 10% of elapsed time actually database transaction
- Production application compiled “/NOOPTIMIZE”
- Recompiled “/OPTIMIZE” and relinked
  - *50% application throughput increase*

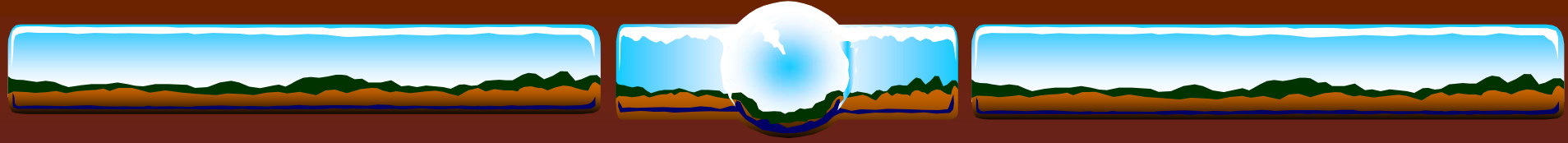


Linker Hints

- **/MAP /FULL /CROSS /SYMBOL_TABLE /DSF**
- both **/DEBUG** & **/NODEBUG**
- **/SECTION_BINDING**
- **LINK /VAX**
  - **VAX 6650 - 153 seconds**
  - **GS1280 – 6 seconds**

Images

- **`$ PIPE -`**
  `SHOW DEV/FILE/NOSYS SYS$SYSDEVICE: | -`
  `SEARCH SYS$INPUT: .EXE;`
- Look for many copies of the same .EXE files
- INSTALL ADD
  - **`/OPEN /SHARE /HEADER [/RESIDENT]`**



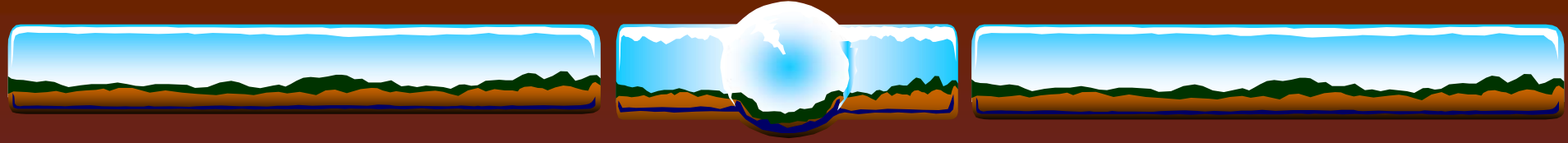
RMS

- **SYSGEN SET RMS_SEQFILE_WBH 1**
- **SET FILE /STATISTICS & MONITOR RMS**
- Use larger buffers & more of them
- Specify FAB/RAB parameters:
  - **RAH / WBH / DFW / SQO / NOSHR / ALQ / DEQ / MBC / MBF**
- RMS After Image Journaling
  - Data protection
  - RMSJNLSNAP freeware tool

Copying 800MB file from disk to disk

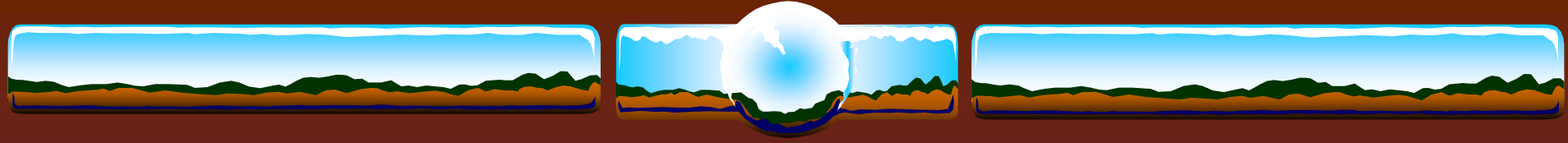
```
Accounting information:  ! VMS V7.3-2
     Buffered I/O count:                61      Peak working set size:          2480
     Direct I/O count:                 26115    Peak virtual size:            168672
     Page faults:                        217    Mounted volumes:                   0
     Charged CPU time:        0 00:00:07.69     Elapsed time:          0 00:02:12.82

 Accounting information:! VMS V7.3-1
     Buffered I/O count:                61      Peak working set size:          2352
     Direct I/O count:                 51758    Peak virtual size:            168672
     Page faults:                        206    Mounted volumes:                   0
     Charged CPU time:        0 00:00:11.22     Elapsed time:          0 00:03:23.67
```

One line change – **RAB$B_MBC=127**



Indexed Files

- **ANALYZE /RMS /FDL** & **RMU /CONVERT**
  - Indexed Files during downtime
- Evaluate larger bucket sizes
- Null Keys?



TCP/IP & DECnet

- TCP/IP V5.4 or later
  - Scaleable Kernel

- Increase default buffer size → reduce BIO
  - **`sysconfig -r inet tcp_mssdflt=1500`**

- **`SET RMS /SYSTEM /NETWORK = 127`**

DECram

- HP VMS product
- Create virtual disk from system memory
- When temp/work files can not be avoided
- Integrated into VMS with V8.2
- May be shadowed with a physical disk
  - Shadow server is smart enough to read from memory

System disk

- Move towards a more “read only” volume
- Move written files off system disk
  - Operator Logs, accounting logs, SYSUAF, NETUAF, RIGHTSLIST, Queue management databases, netserver logs, Rdb monitor logs, etc.
- Remove page/swap files from system disk

Software RAID

- HP VMS product
- Bind local disks into RAID (0 or 5) sets
- “Magically” distribute I/O load among spindles
- Partition RAID arrays into logical units if needed
- Small CPU overhead vs. I/O distribution

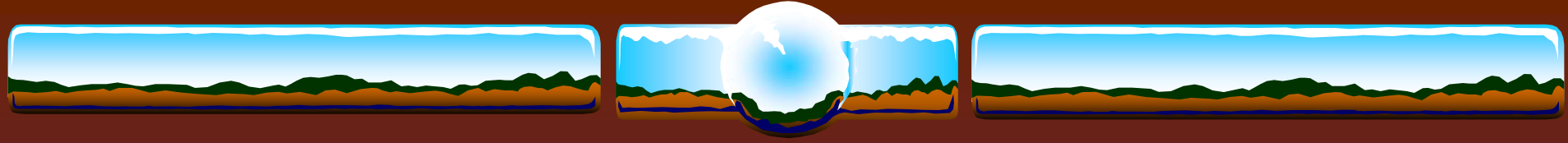
- Or….Use hardware controllers

Disk Volumes

- SET VOLUME
  - /NOHIGHWATER
  - /EXTEND=1024 (+?)
- SET RMS /SYSTEM
  - /BLOCK=64 (?)
  - /BUFF=4 (?)

Backups

- **/CRC /VERIFY**
  - **“The amount of protection that you provide for your data is relative to the amount of value you think your data has”**
- Measure *total* time for restore/recovery
  - including locating, delivering and mounting tapes
- Practice, practice, practice
  - **“There is no need to test the backup procedures… Only the restore procedures!”**



Online Indexed File Backup

- **`CONVERT /SHARE`**
  - Record copy of an indexed file
  - Uncorrupted output file
- Perhaps run prior to online VMS backup for things like SYSUAF, NETUAF, RIGHTSLIST, etc
- Does not address discoordinated updates between files

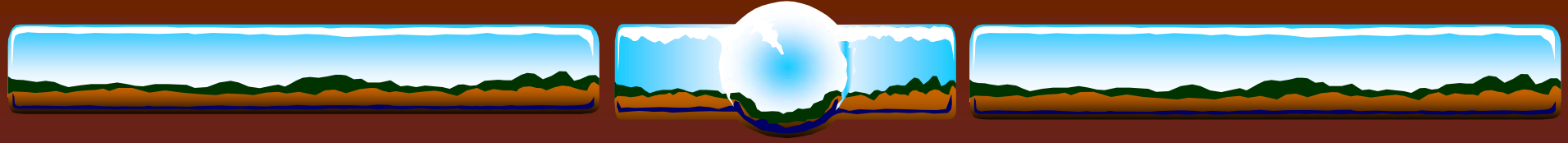
More BACKUP qualifiers

- **/JOURNAL** – so you can find files more easily
- **/TAPE_EXPIRATION** – avoid mistakes
- **/BLOCK_SIZE**=<large> for modern tapes
- **/MEDIA_FORMAT=COMPACTION** where possible
- **/GROUP=100** – perhaps for tapes that do additional data protection on the drive or for disk-based savesets

SPx

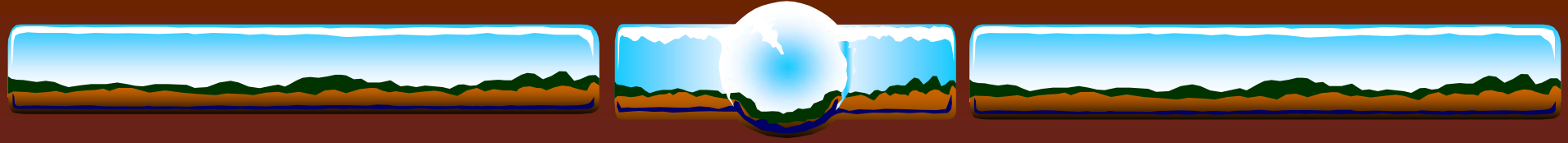
- Quick & Easy Subprocesses to do ‘stuff’

```
$ SPL == “TYPE SYS$SCRATCH:SP.LOG.*”
$ SPN == “SPAWN/NOWAI/NOTIF/NOKEY/INP=NL:”+-
       “/OUTPUT=SYS$SCRATCH:SP.LOG”
$ SPP == “PURGE/LOG SYS$SCRATCH:SP.LOG”
$ SPE == “SEARCH SYS$SCRATCH:SP.LOG.* %”

$ SPN <somedclcommand>
$ SPN <somethingelse>
$ SPN <andsoon>
$ SPE ! Find any possible errors
$ SPL ! Type the log files
$ SPL /TAIL = 10 ! Show end of log files
$ SPP ! Purge old logs
```



Handy SDA Commands

- SDA> SHOW PROC…
  - /IMAGE
  - /LOCKS
  - /CHANNEL
- CLUE
  - SDA> CLUE CALL
  - SDA> CLUE CONFIG
  - SDA> CLUE PROCESS /RECALL
  - SDA> SHOW RESOURCE /CONTENTION



Handy SDA commands

- Finding DCL structures
  - **SDA> READ DCLDEF**
  - **SDA> EXA CTL$AG_CLIDATA+8**
  - **SDA> DEF PRC @.**
  - **SDA> FOR PRC**

Handy SDA commands

- Timer activities
  - **TQE LOAD**
  - **TQE START TRACE**
  - **TQE SHOW TRACE [/SUMMARY]**
- Locking activities
  - **LCK SHOW ACTIVE**
  - **LCK SHO LCK /INT=10/REP=10**

Logical Name Translation

```
SDA> LNM LOAD
SDA> LNM START TRACE
SDA> LNM START COLL /LOGICAL
SDA> LNM SHO COLL
       Count    Logical Name
-----------     ---------------------------
         324    TZ
         218    SYS$SYSROOT
         130    SYS$SHARE
         118    SYS$COMMON
          70    COSI_SRC
          68    SYS$DISK
          60    COSI$CMS
          56    SYS$SPECIFIC
          49    SYS$SYSTEM
          42    TCPIP$INET_DOMAIN
          31    PDEV$COSI
          30    GBL$INS$B3B500D0
SDA> LNM SHO TRACE ...
```

FLT Alignment Fault Tracing

- Ideal is no alignment faults at all !
  - Poor code and unaligned data structures do exist
- Alignment fault summary…
  - **SDA> FLT START TRACE**
  - **SDA> FLT SHOW TRACE /SUMMARY**
  - **flt_summary.txt**
- Alignment fault trace...
  - **SDA> FLT START TRACE**
  - **SDA> FLT SHOW TRACE**
  - **flt_trace.txt**

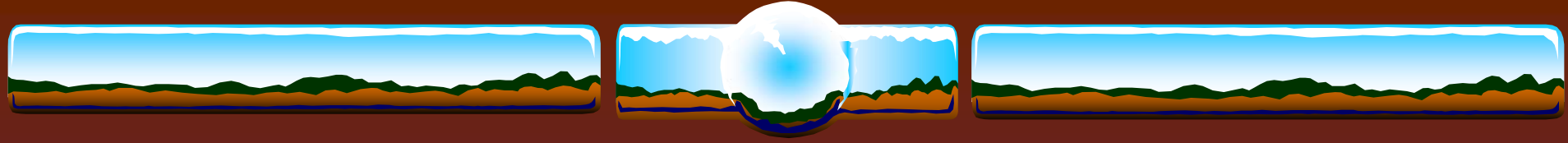
Tools, OpenVMS FreeWare, Hunter Goatley's Freeware – *Don't Leave Home Without…*

- GREP
- AWK
- TECO
- SORT / specification files
- DECRAM
- RZDISK
- ICALCV
- MBX
- ZIP & UNZIP
- LDDRIVER
- DFU
- AlphaPatch
- RMS_TOOLS
- BAT
- Ethereal
  (http://www.ethereal.com/)

VAX Simulators

- Charon VAX - Commercial product from SRI
  - **www.charon-vax.com**
- SIMH - Free VAX Simulator
  - **simh.trailing-edge.com**
- VAX/VMS - faster on a PC or an Alpha?

```
Duo TTA0:> show system
OpenVMS V7.3  on node DUO   2-APR-2003 16:49:08.45  Uptime  0 00:01:15
  Pid    Process Name    State  Pri      I/O       CPU       Page flts  Pages
00000041 SWAPPER         HIB     16        0   0 00:00:00.20          0      0
00000045 CONFIGURE       HIB      8        6   0 00:00:00.09        116    180
     . . .
00000054 njl @ TTA0      CUR      4      172   0 00:00:01.63       1367    467
00000055 RDMS_MONITOR71  LEF     15       18   0 00:00:00.58       1104   1059
Duo TTA0:>
```



QUESTIONS?

"Make your systems scream… Not your users"

- anonymous…