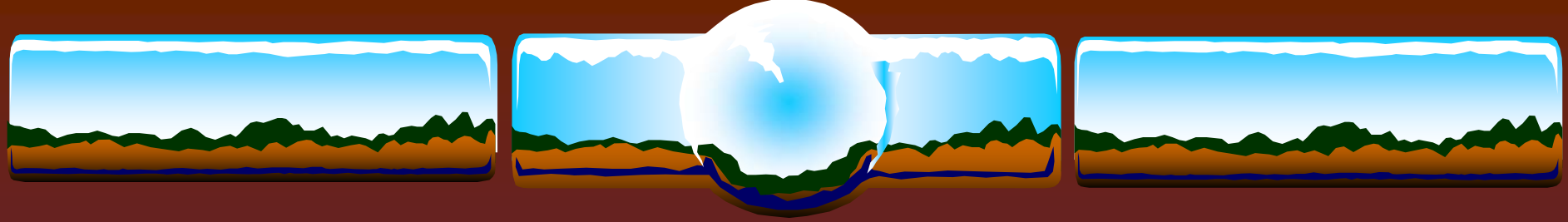


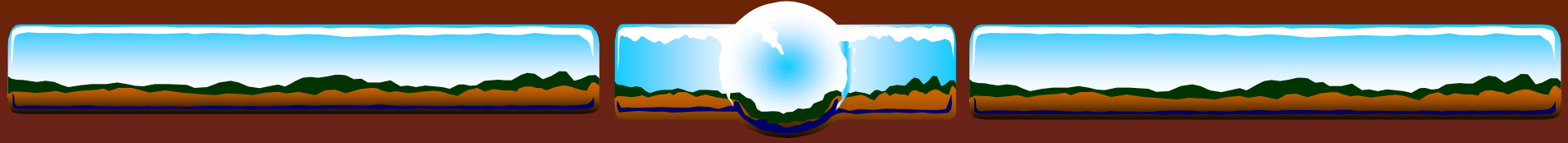
OpenVMS Information Desk

"If at first the idea is not absurd, then there is no hope for it." - *Albert Einstein*

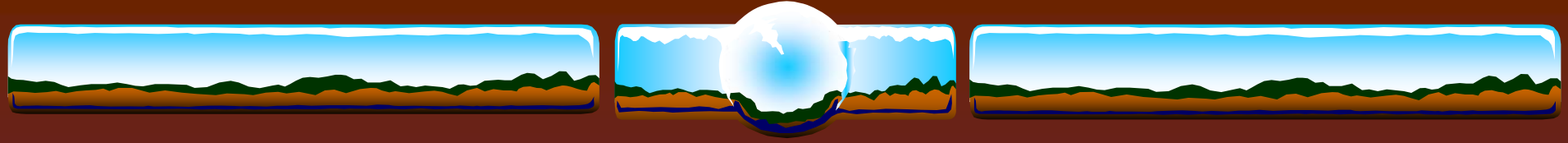
Guy Peleg
HP OpenVMS Engineering

Norman Lastovica
Oracle Rdb Engineering

Update 4 October 2005



The Secrets of Performance

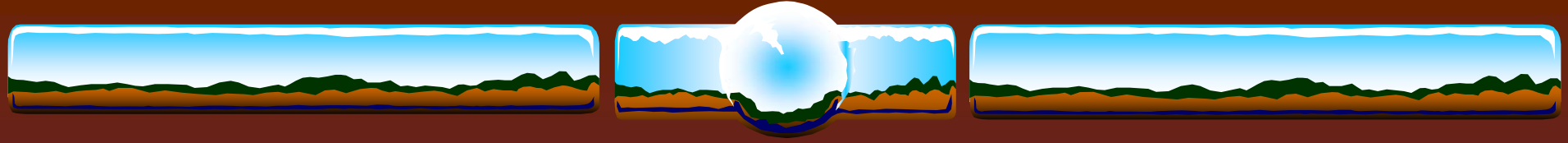


Our Golden Rules

The best performing code is
the code not being executed

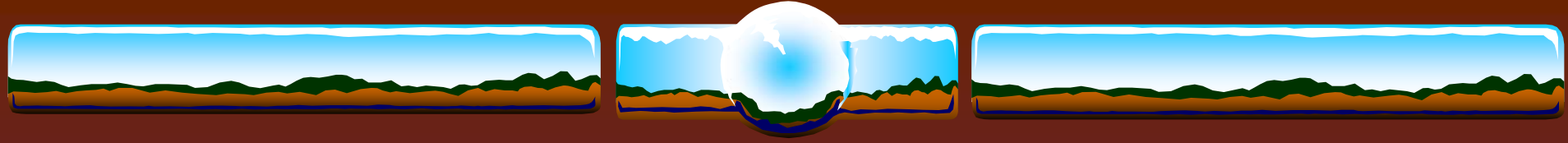
The fastest I/Os are those avoided

Idle CPUs are the fastest CPUs



VMS Versions

- V7.3-1
  - “Required” for > 4 CPUs, Dedicated CPU lock manager, scheduling, fastpath SCSI & FIBER, spinlock contention reductions, TQE improvements, CPU-specific CRTL
- V7.3-2
  - Working set in S2, per mailbox & PCB spinlocks, LAN fastpath, scalable TCPIP kernel
- V8.2
  - IPF (obviously), Fast UCB create/delete, MONITOR enhancements, TCPIP enhancements, large lock value blocks



…..and most important…..

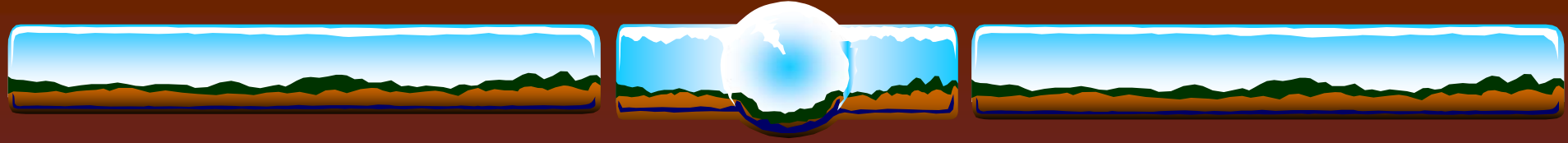
Many new DCL features

Configuration

- Dedicated CPU Lock Manager - Keep it dedicated!
- FastPath
- Path balance
- I/O Adaptors / QBB
- Write-back cache
  - On controllers - Use battery backup
  - On devices
    - Manually/explicitly set flags in disks; sometimes only viable for locally connected SCSI disks; for non-write-critical data

Configuration (cont)

- Wildfire
  - **`SDA> SHOW EXEC /SUMMARY`** - images “sliced”
  - Evaluate RAD-specific processes/global sections
- Marvel
  - RADs likely not a worry
  - CMOS’s configuration suggestion:
    - Connect no IO to duo with primary CPU
    - Connect first IO7 to duo with CPU 2&3
    - Connect second IO7 to duo with CPU 4&5
    - Etc.



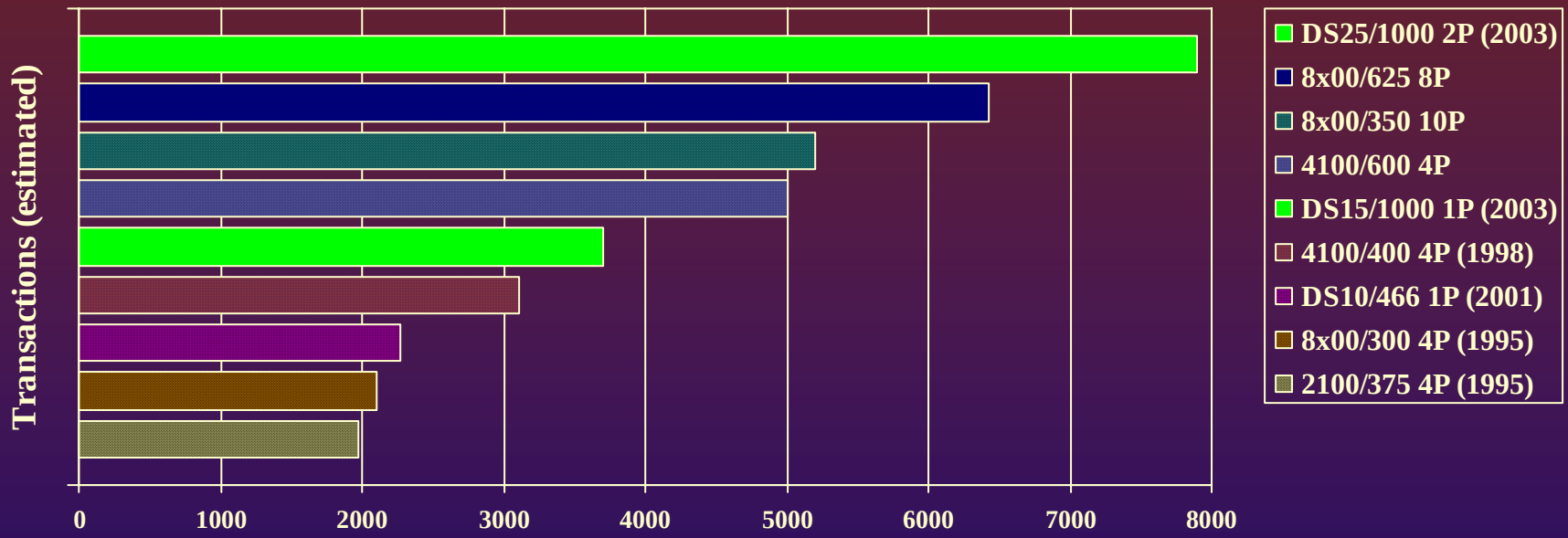
Transition Slide

"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*

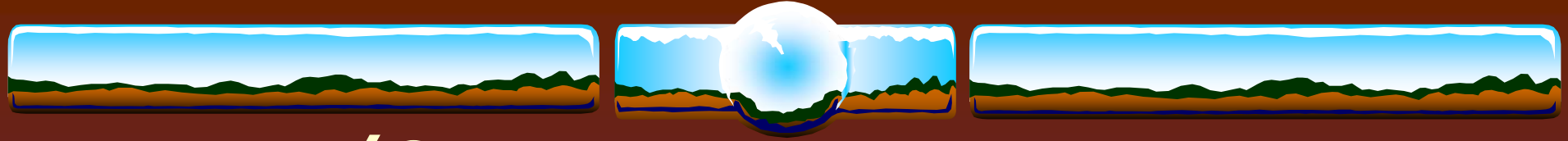
"Application changes have the greatest potential of improving performance" - *Guy Peleg Oct. 15th 12:05*

Slowest system today is faster than fastest was ‘back then’



- ***Significant*** I/O improvements - bus, disk, controller, network
- Often a good consideration
  - Low risk – cost savings?
  - Likely no application changes, recompiles or relinks required

DS10L ~2” tall, uses 1.9A
DS25 ~9” tall, holds 16Gb



/OPTIMIZE=…TUNE=
/ARCHITECTURE=

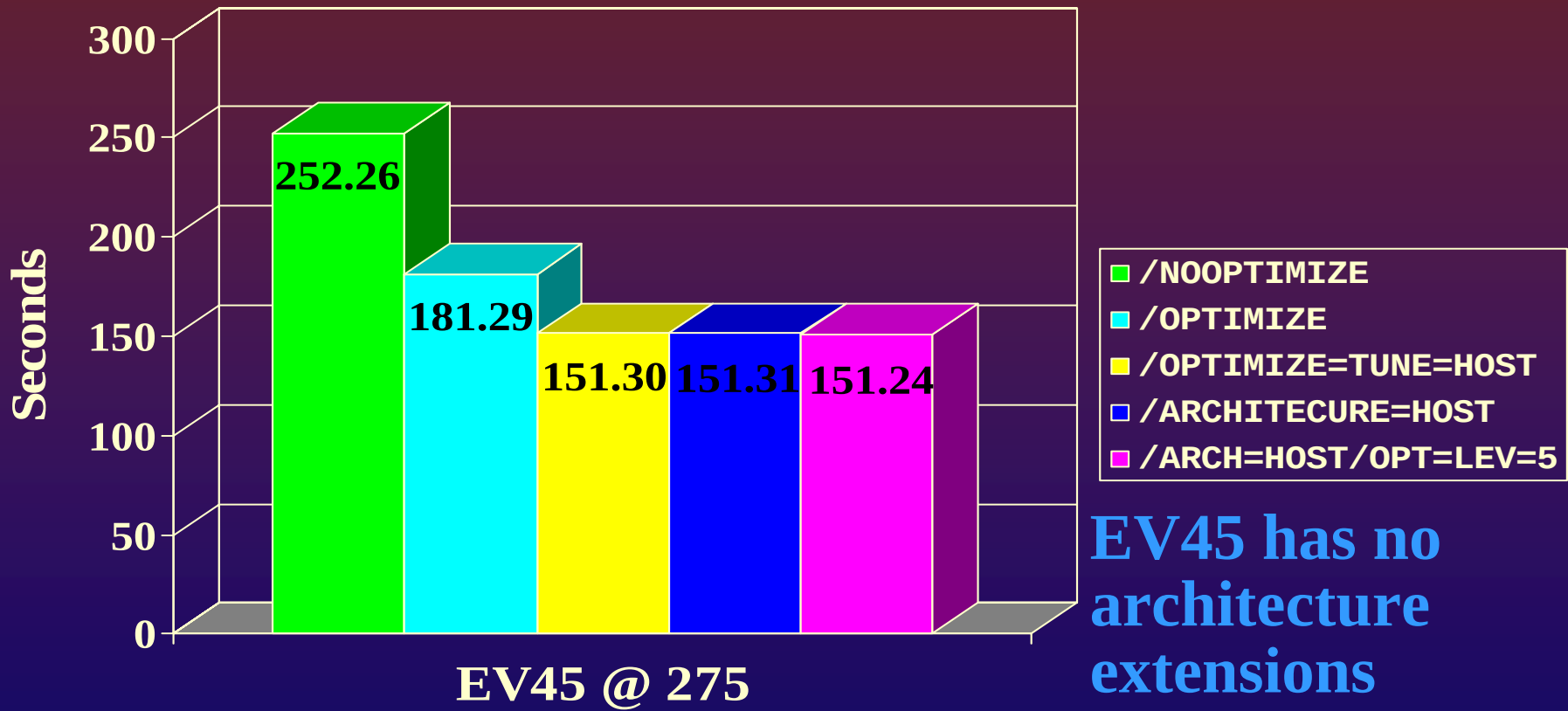
- **TUNE**
  - Code sequences *biased* towards scheduling characteristics of specified processor
  - Can produce code to make CPU run-time decision
- **ARCHITECTURE**
  - Generate code for specified architecture & later
  - Optimal instruction scheduling & use of available instructions

Prime Numbers Test

- Find first 1,000,000 prime numbers

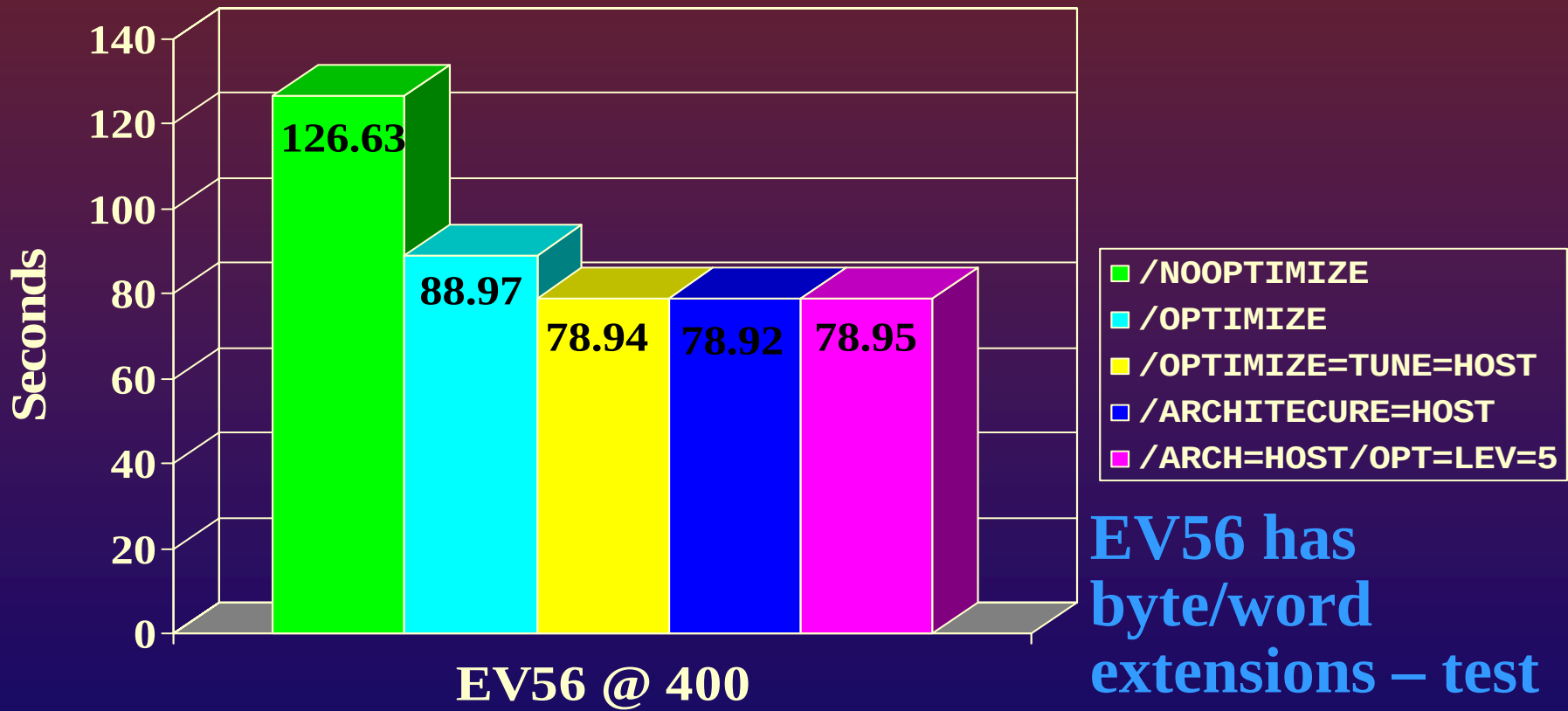
```
        primes(1) = 3
        hi_prime = 3
        hi_prime_index = 1
        hi_prime_divisor_index = 1
        do 100 i = 5,2000000000,2
        if (primes(hi_prime_divisor_index)**2 .lt. i)
            hi_prime_divisor_index = hi_prime_divisor_index + 1
        do 20 j = 1, hi_prime_divisor_index
        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



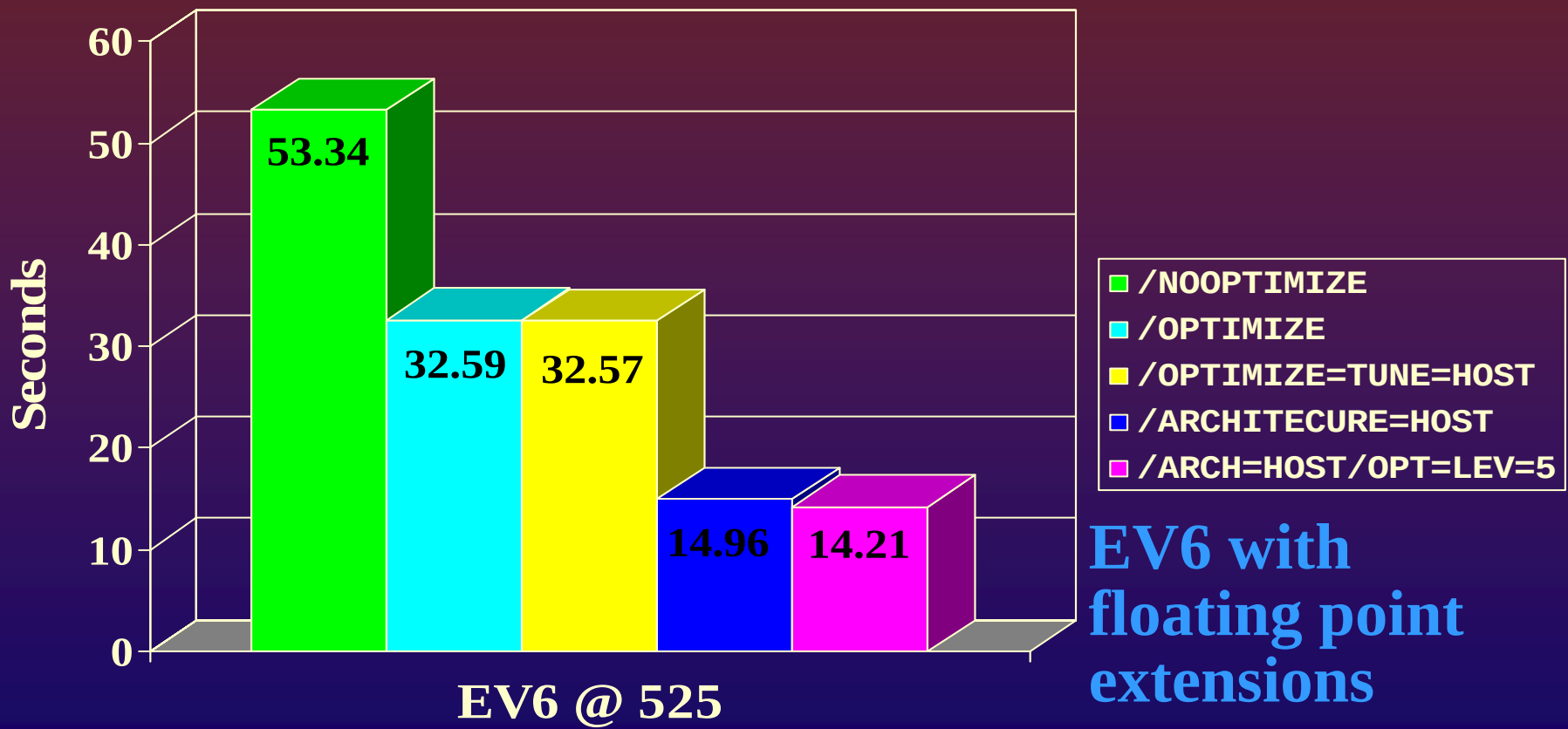
EV45 has no architecture extensions

Generating Primes
AlphaServer 4100 5/400



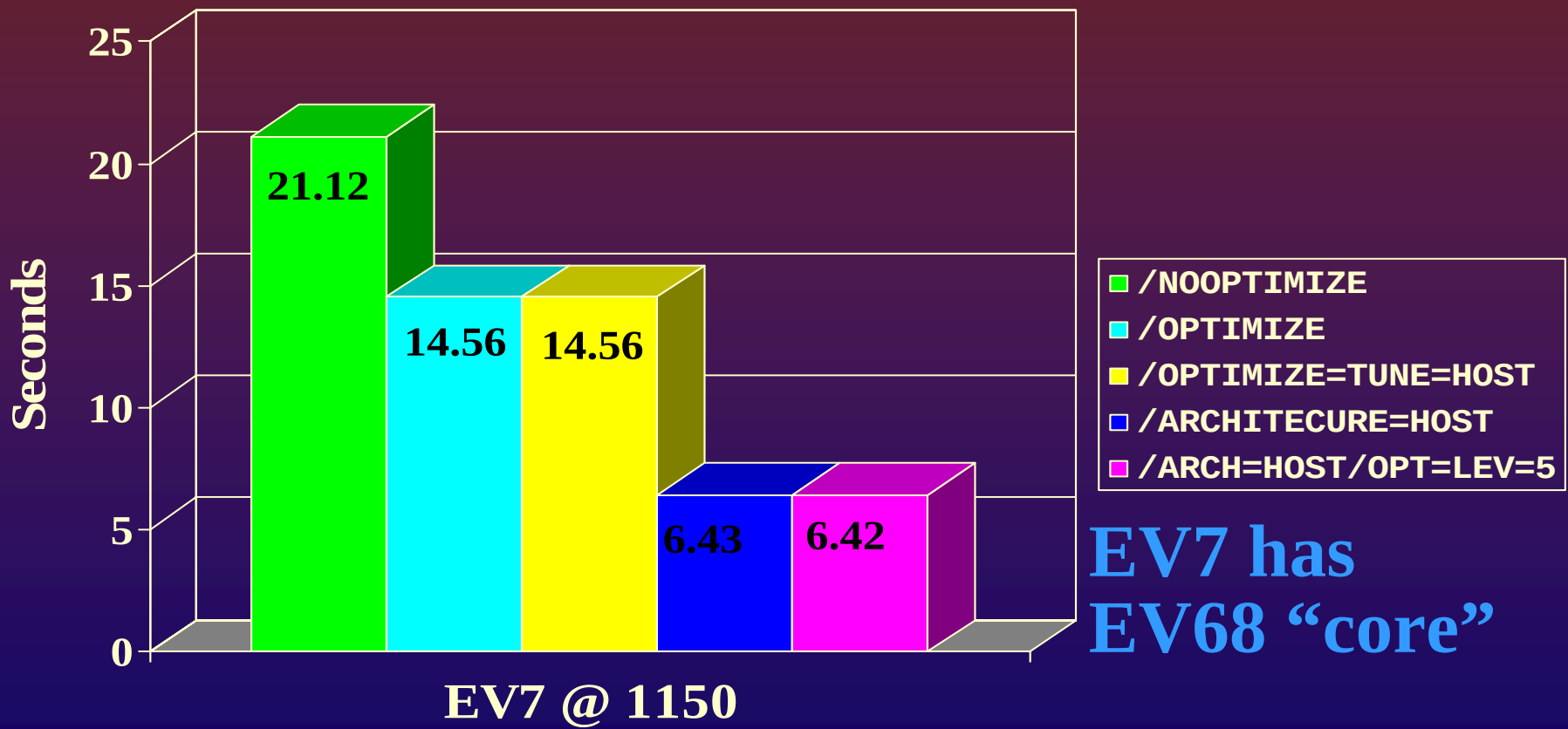
EV56 has byte/word extensions – test does not use them

Generating Primes
AlphaServer GS140 6/525



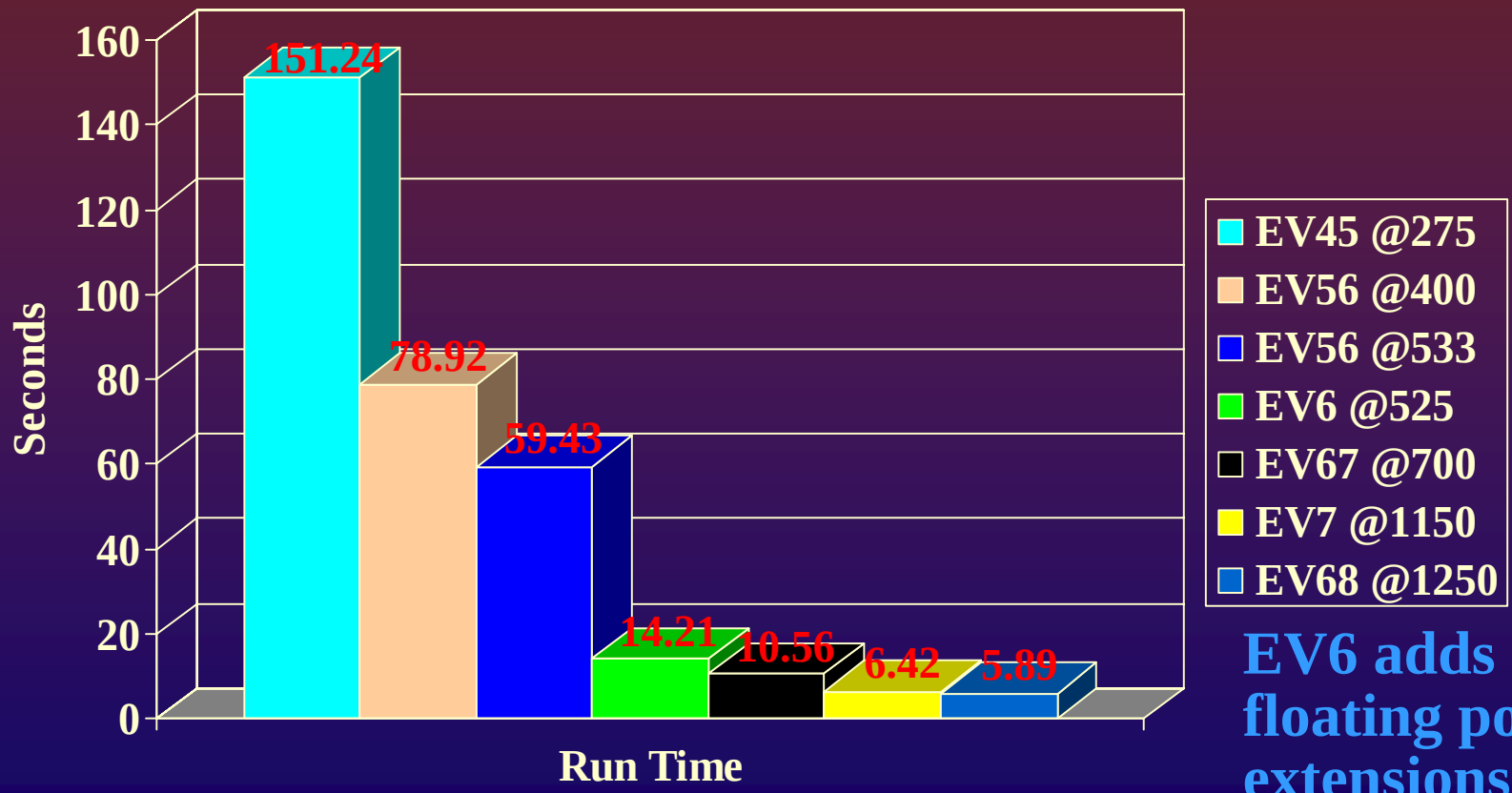
EV6 with floating point extensions

Generating Primes
GS1280 7/1150



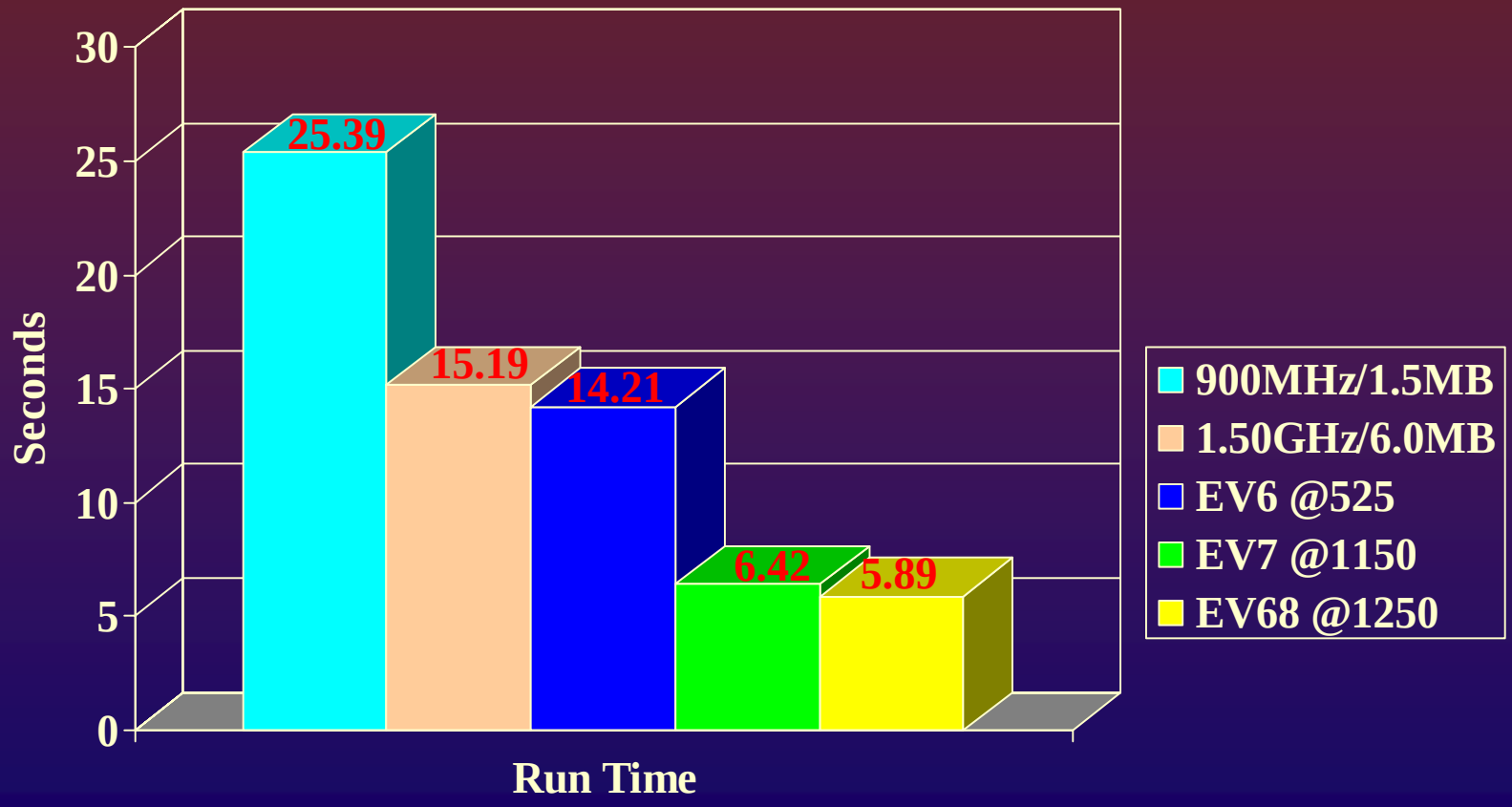
EV7 has EV68 “core”

Generating Primes… Comparing the Machines

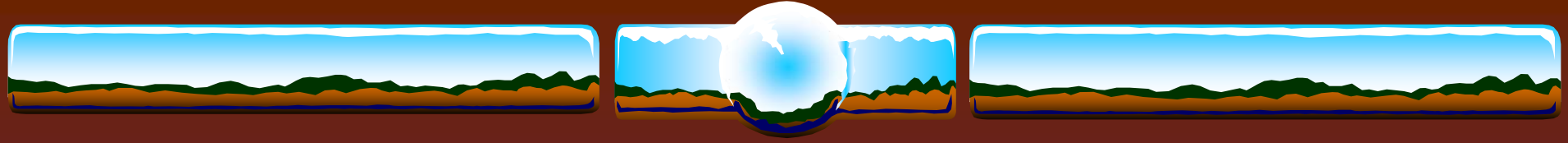


EV6 adds floating point extensions & quad-issue

Comparing the Machines
PRIME.FOR

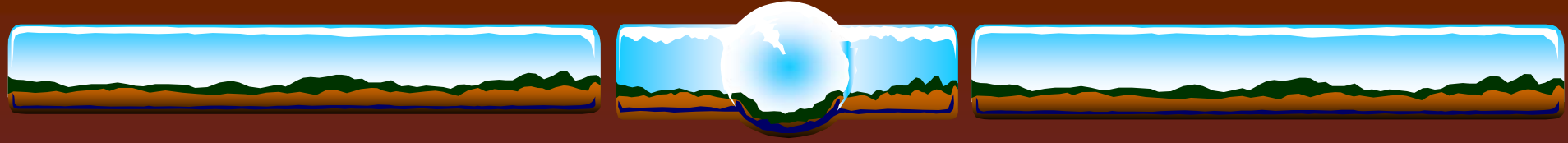



```
FORT /OPT=LEVEL=5 /ARCH=HOST PRIME.FOR
```



Real-life Example

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



Initializing Structures
Which is fastest/efficient?

- Initializing structures in BLISS....

.....Wait a second, how many people around here use BLISS….☺

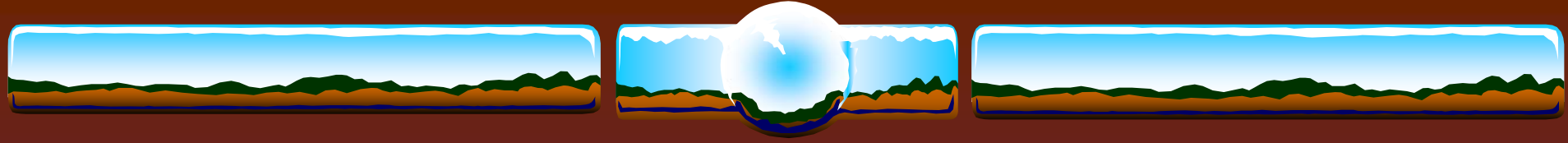
……. Let’s try again…..

Initializing Structures
Which is fastest/efficient?

```
void foo1 (){
    char array[512]={0};
    printf("array=%x",&array);}

void foo2 (){
    char array[512];
    for (int i=0;i<512;i++) array[i]=0;
    printf("array=%x",&array);}

void foo3 (){
    char array[512];
    memset (array, 0, sizeof(array));
    printf("array=%x",&array);}
```

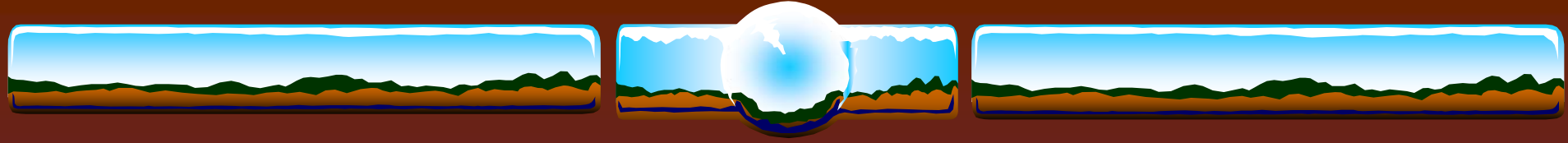
setjmp

```
main(char **av, int ac)
{ time_t tm = time(0);
  int i, env, nosetjmp = 0;

  if ((ac == 2) && (*av[1] == '-')) {
    printf("No setjmp\n");
    nosetjmp = 1; }

  lib$init_timer();

  for (i = 0; i++ < 1000000;) {
    if (nosetjmp) env = i;
    else {
      env = setjmp(g_jmpbuf);
      if (env) printf("Jumped\n"); } }

  lib$show_timer(); }
```



setjmp

- Takes 45 seconds to execute this program on 8P/8C Superdome (1.5GHZ)
- Compiled with **`/define=__FAST_SETJMP`** program takes only 0.05 seconds



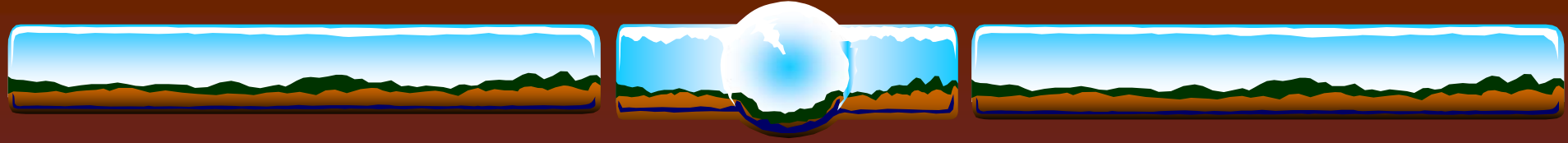
Performance and Coverage Analyzer

- Use PCA with your applications!
- Find where time is being spent – focus first on those areas
- Identify I/O, System Service, Alignment Faults, PC sampling, etc.

Linker

- **/DSF**
- **/SYMBOL_TABLE**
- **/MAP /FULL /CROSS**
- **/SECTION_BINDING**

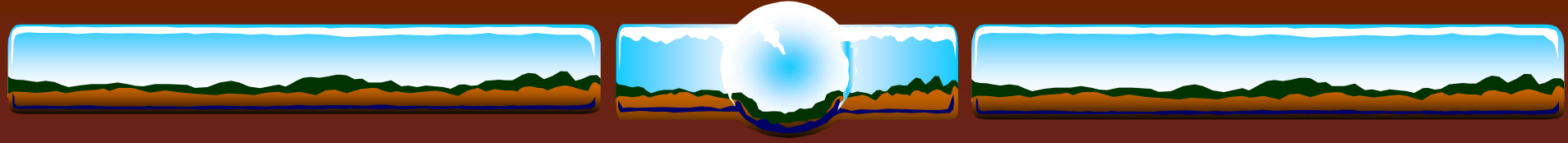
- **LINK /VAX**
  - VAX 6650 - 153 seconds
  - GS1280 – 6 seconds



Images

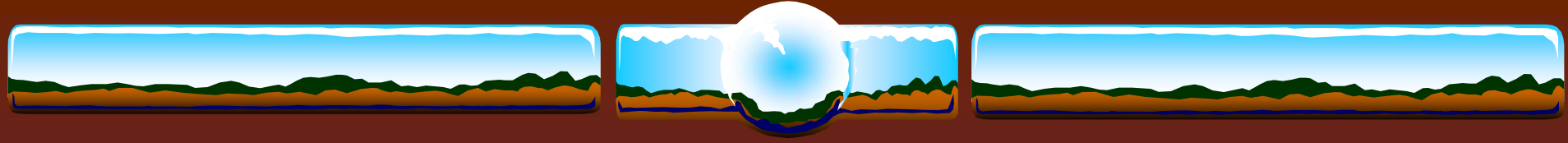
- $ PIPE –
  SHOW DEV/FILE/NOSYS SYS$SYSDEVICE: | -
  SEARCH SYS$INPUT: .EXE;

- INSTALL ADD …
  - /OPEN /SHARE /HEADER [/RESIDENT]



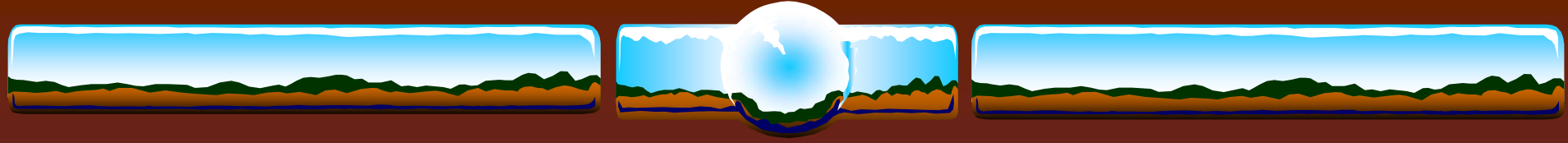
RMS

- **SYSGEN> SET RMS_SEQFILE_WBH 1**
- **SET FILE /STATISTICS**
  - **MONITOR RMS**
- After Image Journaling for data protection
  - RMSJNLSNAP freeware tool



RMS

- Use larger buffers & more of ‘em
- FAB/RAB parameters:
  - **ASY, RAH, WBH, DFW, SQO**
  - **ALQ & DEQ**
  - **MBC & MBF**
  - **NOSHR, NQL, NLK**
- **SET RMS …**
  - **/SYSTEM**
  - **/BUFFER_COUNT=n**
  - **/BLOCK_COUNT=n**



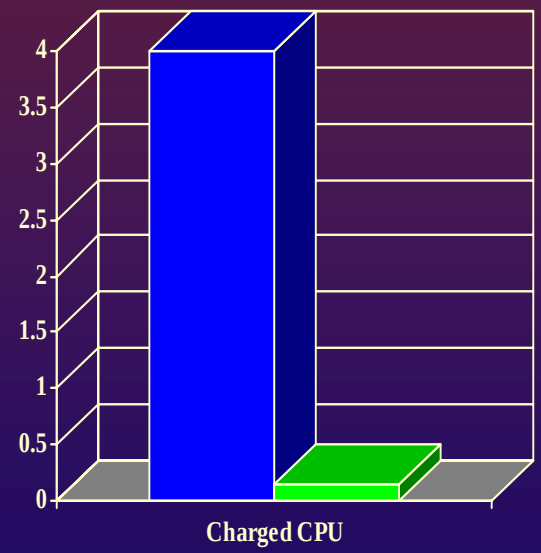
Copying 800MB file disk to disk

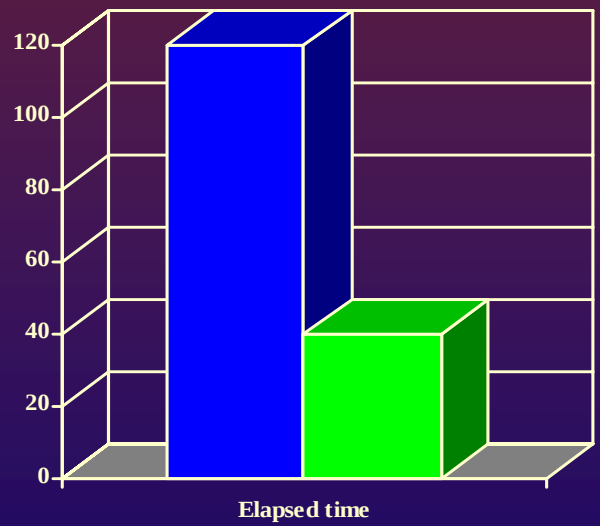
```
Accounting information:     ! VMS V7.3-1
  Buffered I/O count:                 61     Peak working set size:        2352
  Direct I/O count:                51758     Peak virtual size:          168672
  Page faults:                       206     Mounted volumes:                 0
  Charged CPU time: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

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
```

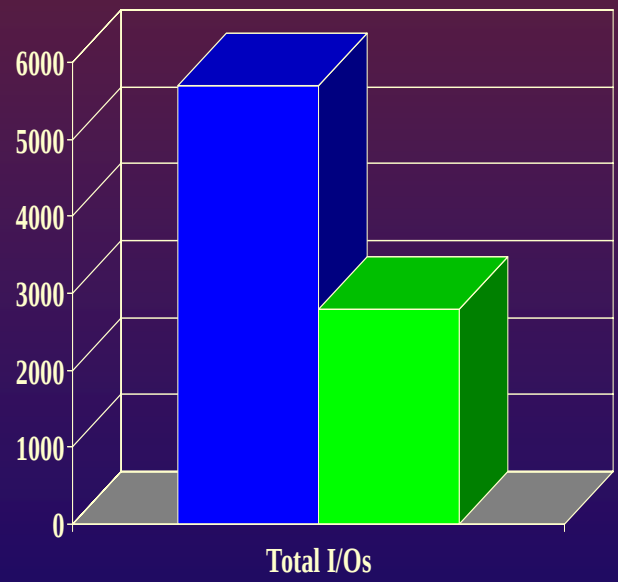
One line change – **RAB$B_MBC=127**

Sometimes….redesign is the answer

Copy 170MB file disk to disk







V7.3-2 image

Nemo Prototype

MACRO-32 BBS->BLBS

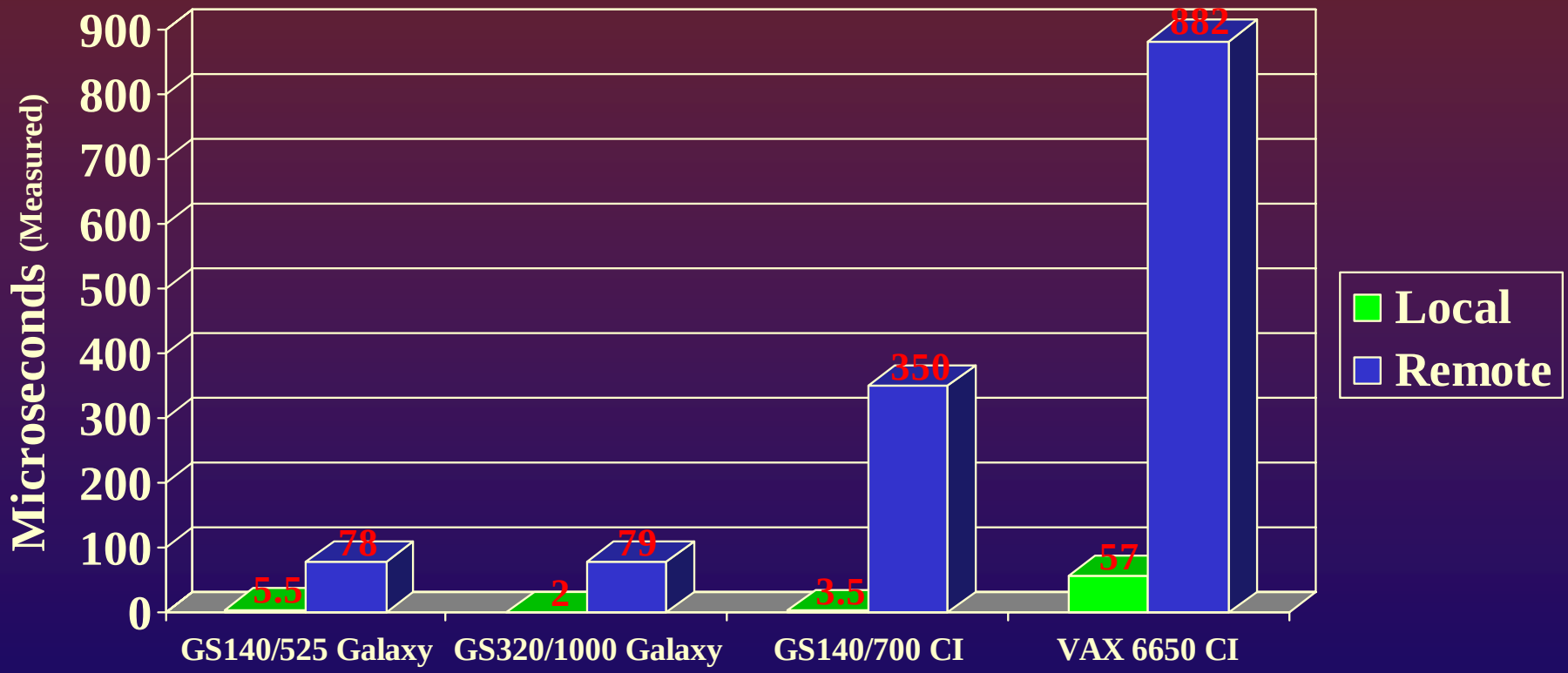
- BBS can be quite expensive
  - Simple change improved elapsed time of executing a DCL loop by 1%

```
MTMERU> diff symbol.mar; ;1/par
------------------------------------------------------------------------------
File WORK2:[PELEG.TOPAZ]SYMBOL.MAR;3   |  File WORK2:[PELEG.TOPAZ]SYMBOL.MAR;1
------------------------------ 499 ---------------------------------- 499 --------------
 BLBS  WRK_L_MED(R10),95$              |  BBS   WRK_V_NOMEDDLE,WRK_W_FLAGS2(R10),95$
------------------------------ 719 ---------------------------------- 718 --------------
 BLBS  WRK_L_MED(R10),25$              |  BBS  #WRK_V_NOMEDDLE,WRK_W_FLAGS2(R10),25$
------------------------------------------------------------------------------
```

Locking

- Balance LOCKDIRWT based on CPU power & workload
- **MIN_CLUSTER_CREDITS=128**
  - For clusters with big/fast machines
- Consider **DEADLOCK_WAIT=1**
  - If deadlocks getting detected – beware “empty” searches
  - It ain’t 1982 any longer

Local Locks are Faster Locks



EVA/XP Storage

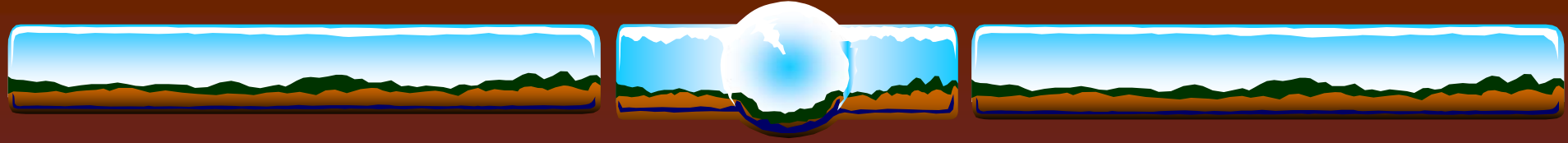
- Initialize disks with cluster size multiple of 4
- Perform sequential write I/O…
  - Multiple of 4 block transfers
  - Starting on multiple of 4 block VBN
  - COPY/BLOCK_SIZE (V8.2)
  - Avoid excessive async sequential access I/O queues
  - Throttle BACKUP quotas

XP storage

- Works best if 8 I/Os per LUN are presented by the host.
- OpenVMS has 3 methods that can help
  - Lower values for DIOLM and PQL_MDIOLM
  - BACKUP/QUOTA=DIOLM
    - Available now to be introduced in future ECO kit
  - WWID throttle IO descriptor to limit the total number of I/Os per FC port
    - V7.3-2 FIBRE_SCSI-V0400 and later
    - SDA> FC SET WTID /WWID=target_wwid /CAP=cap_value

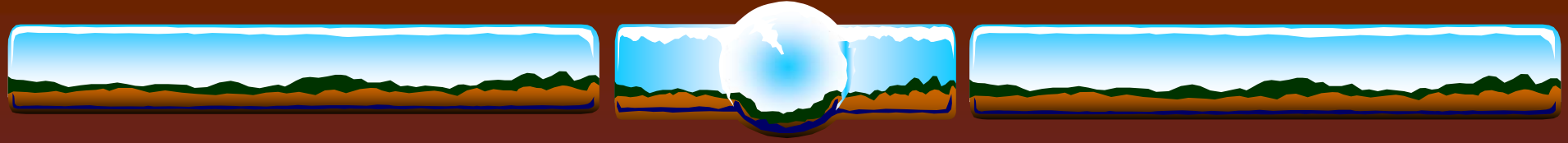
MSCP Disk Serving

- Alpha & I64 MSCP server does not do dynamic balancing
  - **SET PREFERRED /HOST=<node>/FORCE <dev>**
- **MSCP_LOAD** in ‘mixed size’ cluster
  - 2 for “small” Alpha/IPF servers
  - 1 for “big” ones (uses default of 340 on Alpha & IPF)
- **MSCP_CREDITS >=** 64 for busy/big servers
- **MSCP_BUFFER** >= 2048
  - **127 * MSCP_CREDITS** when using host-based shadowing



The Tech Commandments

- *Thou shalt backup, backup, BACKUP!*
- *Thou shalt not make thy password be "password".*
- *Thou shalt not adopt early or install thy version 1.0.*
- *Thou shalt not steal thy neighbor's bandwidth.*
- *Thou shalt not covet thy neighbor's toys. Instead, buy a newer model.*
- *Thou shalt not open unknown email attachments nor reply to SPAM.*
- *Thou shalt use a firewall.*
- *Remember the Slackith days. Six days thou shalt slack and do all thy surfing.*
- *Don't be Evil.*
- *Thou shalt not curse at thy computer when thy problem lies with its user.*



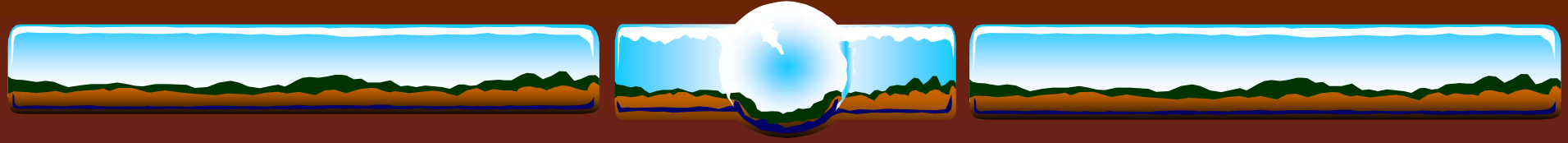
Logical Names

- **DEFINE/SYSTEM/EXECUTIVE DECC$ENABLE_GETENV_CACHE 1**

- **LNMSHASHTBL >= 8192**

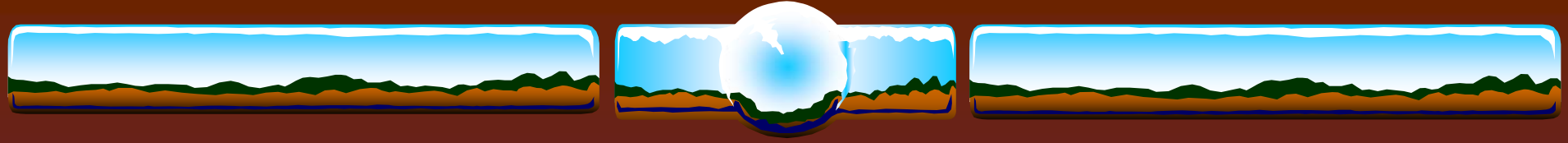
LNM - Logical Name Translation

```
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 ...
```



Indexed Files

- **ANALYZE /RMS /FDL**
  EDIT /FDL [/SCRIPT=OPTIMIZE]
  RMU /CONVERT
  SET FILE /STATISTICS
- Consider larger bucket sizes
- “Null Key” can help you
- Long duplicate chains can kill performance
- Global buffers!



TCP/IP & DECnet

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

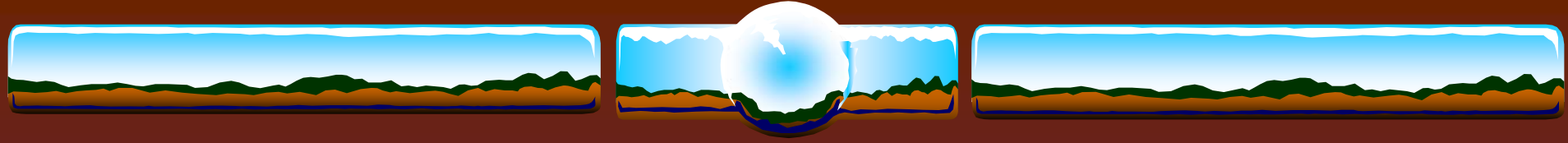
- Increase default buffer size → reduce BIO
  - **`sysconfig -r inet tcp_mssdflt=1500`**

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

POOL

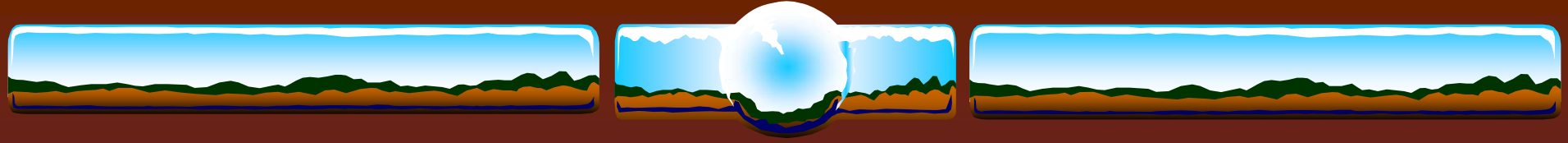
- **NPAG_GENTLE=NPAG_AGGRESSIVE=100** to disable pool reclamation – Current VMS default

- Leave **NPAG_GENTLE** and **NPAG_AGGRESSIVE** out of MODPARAMS



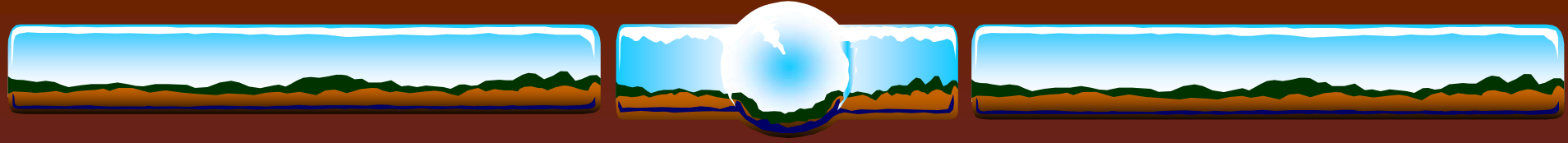
Application Temporary Files

- Frequently create/delete small temp files?
  - Consider caching in virtual memory instead
    - “Spill” to disk file if needed after some threshold (1mb?)
- Don’t be afraid of P2 virtual address space
  - Keep an eye out for excessive page faulting



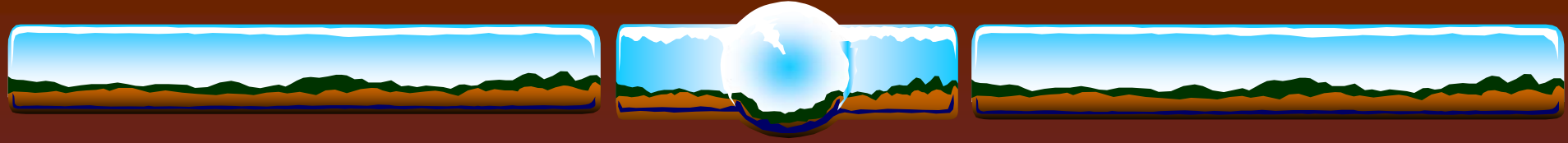
Large Sequential Files

- Rarely read?
  - Create in a directory marked **`/CACHE=NOCACHE`**
- Perhaps for…
  - Backup savesets, unload data, log files, .MAP files, etc
- Avoids polluting XFC cache



Global Sections

- Memory resident
  - Shared page tables
  - Granularity hints
- P2 virtual address space
- Aligned data structures
- Per-RAD sections on Wildfire



XFC

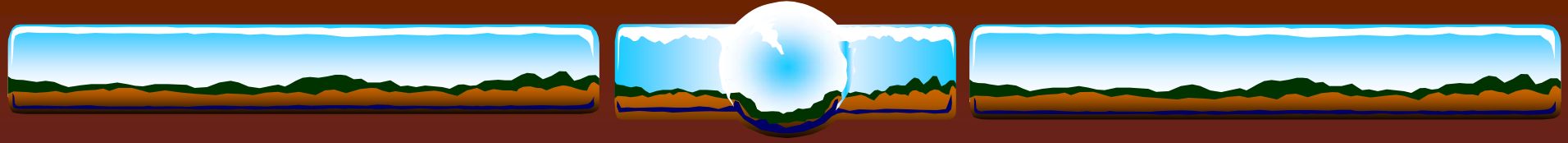
- It isn’t 1980 any longer…
  - Historically I/O sizes maxed at 127 blocks.
  - Today, utilities are doing I/O up to 256 blocks at a time
- Consider setting VCC_MAX_IO_SIZE to 256

DECram

- Create virtual disk from system memory
- When temp/work files can not be avoided
- Integrated with VMS V8.2
- May be shadowed with physical disk
  - Shadowing smart enough to read from memory

System disk

- Move towards more “read only”
- Move written files off system disk
  - Operator Logs, accounting logs, SYSUAF, NETUAF, RIGHTSLIST, Queue management databases, netserver logs, software log files, page/swap files

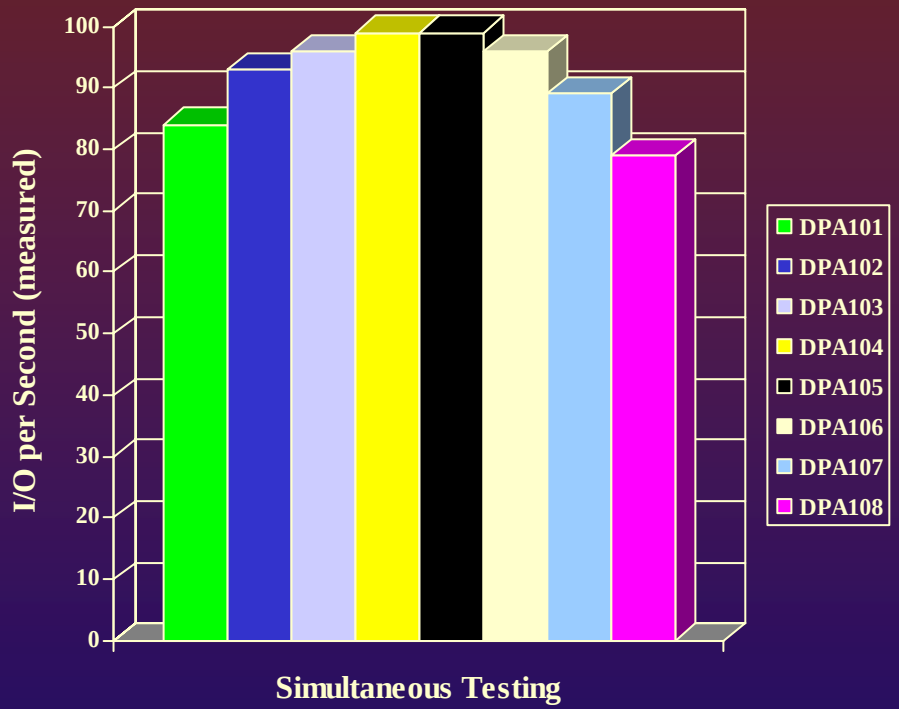


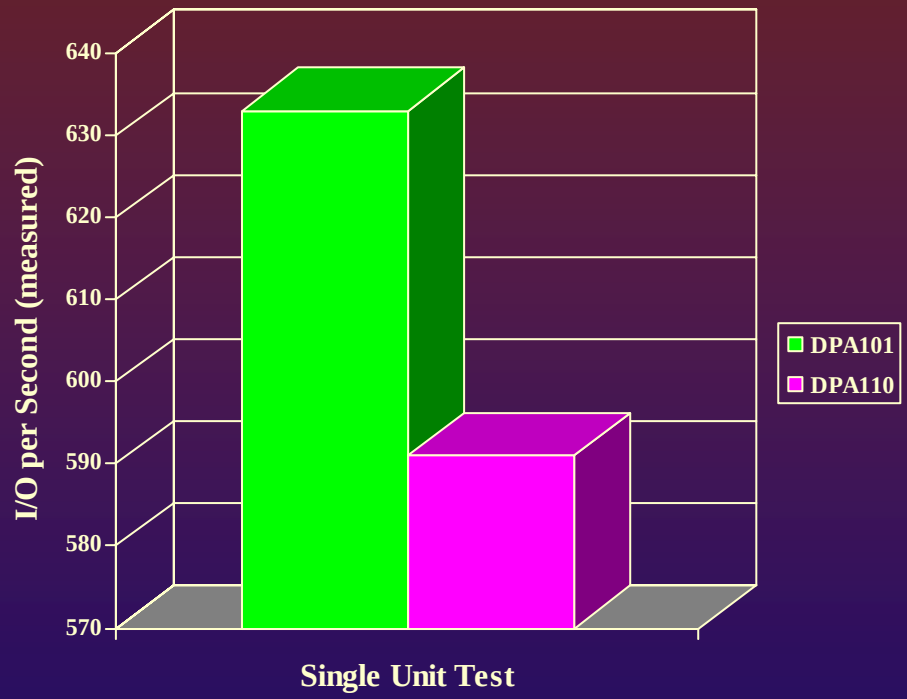
Software RAID

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

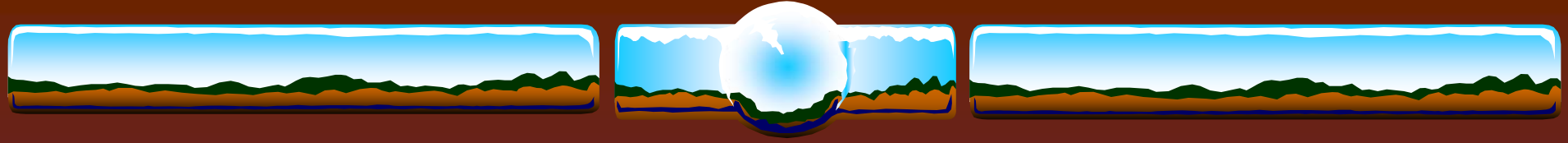
- Or….Use hardware controllers

Performance Indicators



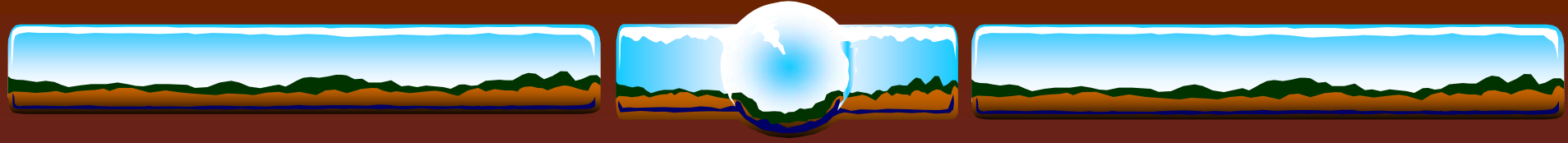


- 10 Virtual units constructed on 6 RZ28 disks connected to HSJ50
- 5 block random reads; Queue depth of 16
- "Machines take me by surprise with great frequency" – *Alan Turing*



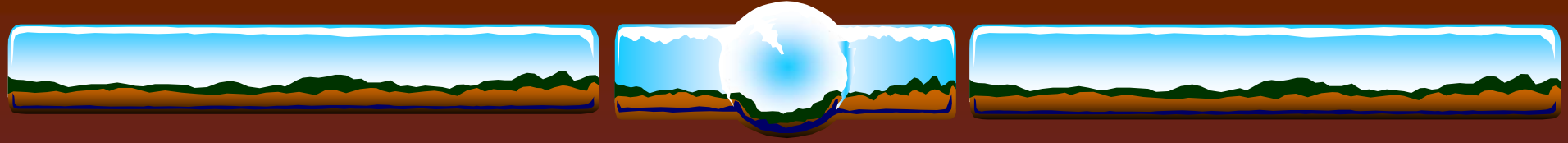
LDDRIVER

- Create virtual disks out of container files
  - Want to use ODS-5
  - Test/debug
- Trace I/O requests
  - Size & LBN
  - Excellent for uncovering performance activity



Disk Volumes

- **SET VOLUME**
  - **/NOHIGHWATER**
  - **/EXTEND=big?**
  - **/CLUSTER=<multiple-of-4>**
  - **/LIMIT**

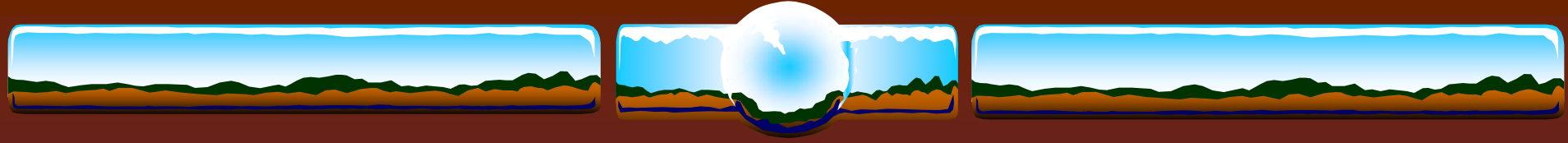


Backups

"The amount of protection that you provide for your data is relative to the amount of value you think your data has"

"There is no need to test backup procedures…
Only the restore procedures!"

"Honest criticism is hard to take,
particularly from a relative, a friend, an acquaintance,
or a stranger" – *Franklin P. Jones*



BACKUP Performance?

- Focus on *total* **restore & recovery** performance…
  - Locate media, transport media, mount it, etc
  - Zero TPS when the system is down

However…if you do care about performance…

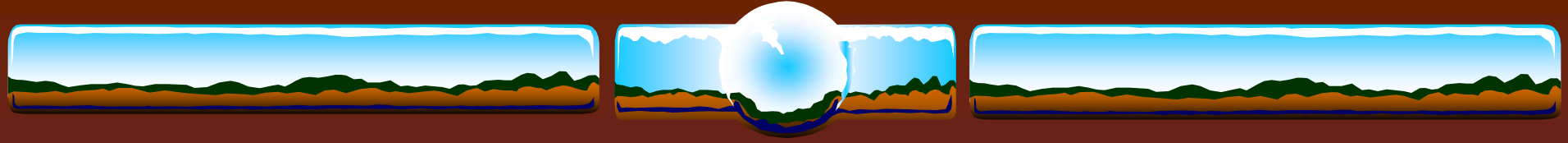
BACKUP Performance

- Backup of 12GB to SDLT on FC
  - **DIOLM=32767, BIOLM=32767, WSDEF=WSQUOTA=WSEXTENT=3000000**
  - **DIOLM=150, BIOLM=150, WSDEF=10000,WSQUOTA=20000,WSEXTENT=30000**
- Which is faster? Remember that sometimes more is less!

BACKUP Performance

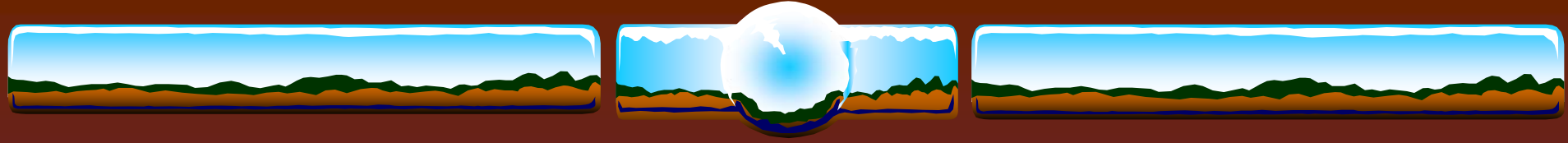
(Do we really need to use media=comp)

- Without compaction SDLT320 can…
  - write 16MB/sec & read 14.89MB/sec
- With compaction doing 8KB I/Os
  - write 28.96MB/sec & read 16.65MB/sec
- With compaction doing 63KB I/Os
  - write 49.54MB/sec & read 32.85MB/sec



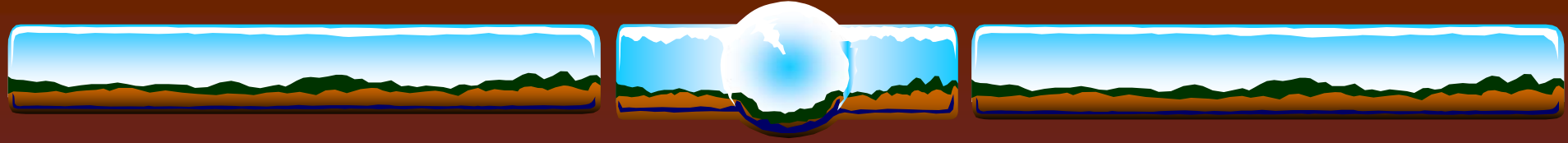
More BACKUP

- Starting with V8.2, BACKUP/PHYSICAL no longer requires identical sized input & output volumes
- Consider using /GROUP=100 for disk-based savesets to reduce size
- Beware John The Ripper - Protect your SYSUAF
  - **http://www.openwall.com/john/**



Online Indexed File Backup

- **`CONVERT /SHARE`**
  - Record copy of indexed file
  - Uncorrupted output file
- Run prior to online VMS backup for things like SYSUAF, NETUAF, RIGHTSLIST, etc.
- Discoordinated updates among files potential issue

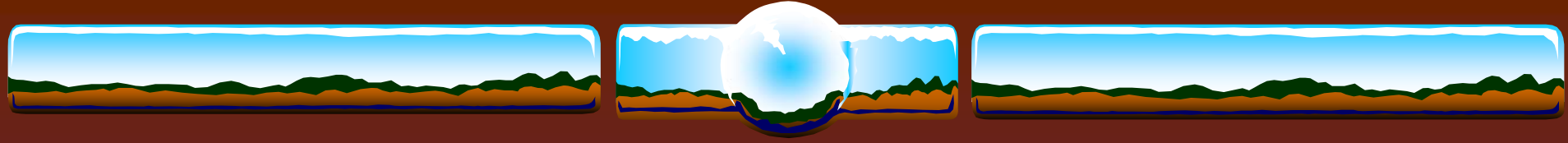


DELETE

- **DELETE /LOG** & **PURGE /LOG** require that files be opened prior to being deleted!
  - Can dramatically increase I/O
- Deleting an entire directory tree? Try DFU.
  - V8.3 ?????

SORTing

- HYPERSORT
  - Multi-threaded
  - Contact HP support for latest update
- Spread work files among disks/controllers/adaptors
  - Apart from input/output disks
  - No problem to have input and output on same disk
- Specification files are very powerful

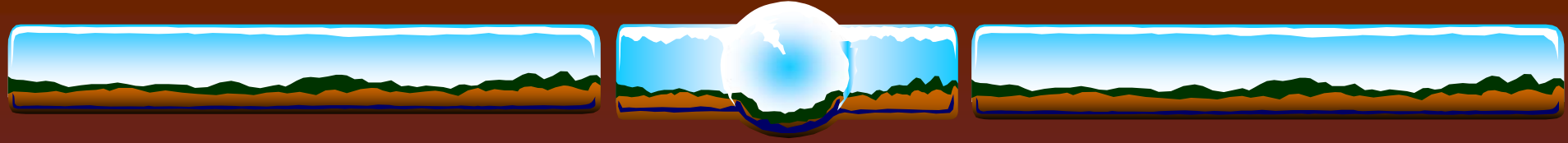


Virtual Terminals

- Avoid process deletion at network disconnect (PC crash?)

Add to system startup:

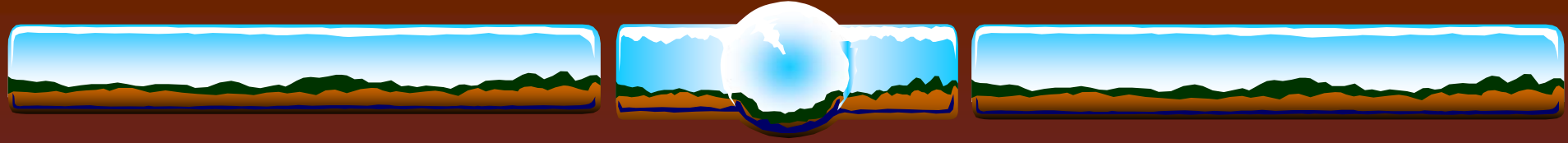
```
$ ! ENABLE VIRTUAL TERMINALS
$ MCR SYSMAN IO CONNECT /NOADAPT VTA0 -
    /DRIVER=SYS$LOADABLE_IMAGES:SYS$TTDRIVER
$ DEFINE/SYSTEM/EXECUTIVE TCPIP$TELNET_VTA TRUE
```



SPx

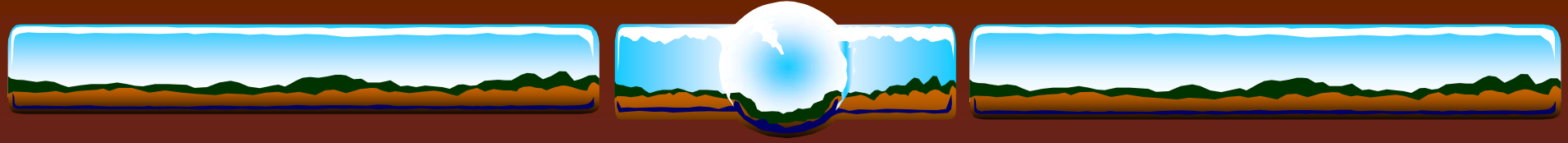
- Subprocesses to do ‘stuff’ and not tie up a terminal
- Similar tricks with batch jobs possible

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

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



Handy SDA Commands

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

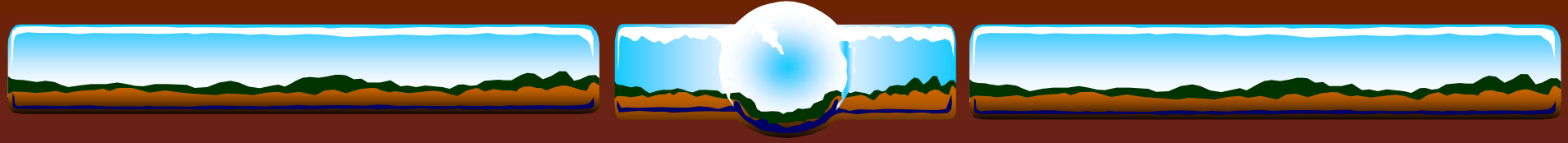


Handy SDA commands

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

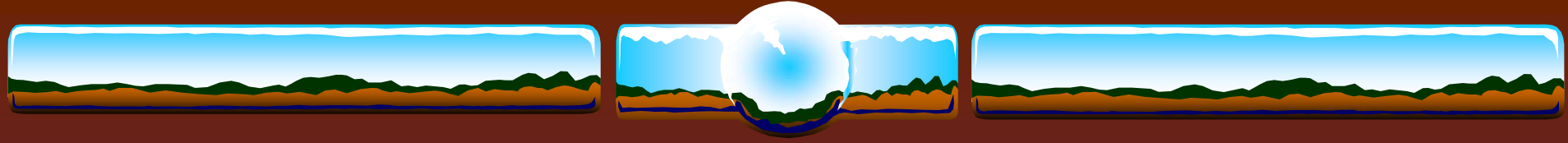
Handy SDA commands

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



FLT - Alignment Fault Tracing

- Ideal is no alignment faults at all!
  - Poor code & unaligned data structures do exist
  - Faults on I64 vastly slower than on Alpha
- 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**

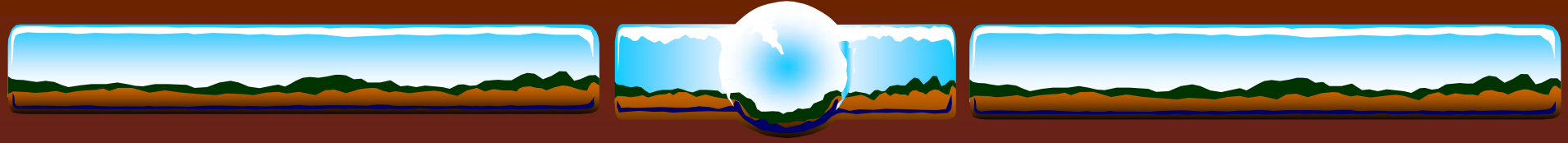


PRF - Processor Counters

- Exceedingly powerful and easy to use
- Link your images /TRACEBACK /SECTION_BINDING and install /RESIDENT

PRF - Example

```
SDA> PRF LOAD
SDA> PRF START PC /MODE=E
SDA> PRF START COLL
SDA> PRF SHO COLL

Start VA          End VA            Image                    Count    Percent
----------------- ----------------- -------------    -----------  --------
00000000.00000000 00000000.7ADBC000 Process Space            30152    63.47%
FFFFF802.07000000 FFFFF802.07014000 Kernel Promote VA          261     0.55%
FFFFFFFF.80000000 FFFFFFFF.800000FF SYS$PUBLIC_VECTORS          34     0.07%
FFFFFFFF.80000100 FFFFFFFF.800111FF SYS$BASE_IMAGE            3548     7.47%
FFFFFFFF.8007DE00 FFFFFFFF.8007EEFF RMS                          2     0.00%
FFFFFFFF.80080000 FFFFFFFF.801AE6FF SYSTEM_PRIMITIVES_MIN     8065    16.98%
FFFFFFFF.8030E200 FFFFFFFF.80402DFF EXCEPTION                 1372     2.89%
FFFFFFFF.80402E00 FFFFFFFF.804E0EFF IO_ROUTINES                 29     0.06%
FFFFFFFF.804F0500 FFFFFFFF.806207FF PROCESS_MANAGEMENT         275     0.58%
FFFFFFFF.80620800 FFFFFFFF.807241FF SYS$VM                     177     0.37%
FFFFFFFF.80727E00 FFFFFFFF.8077B8FF LOCKING                     65     0.14%
```

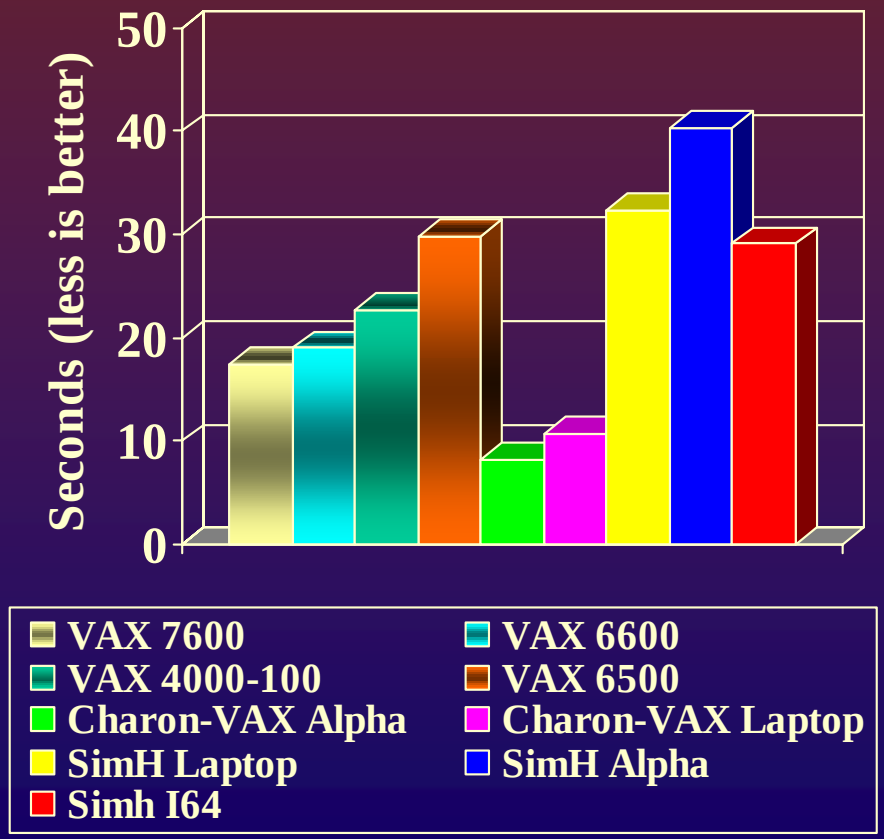


VAX Simulators

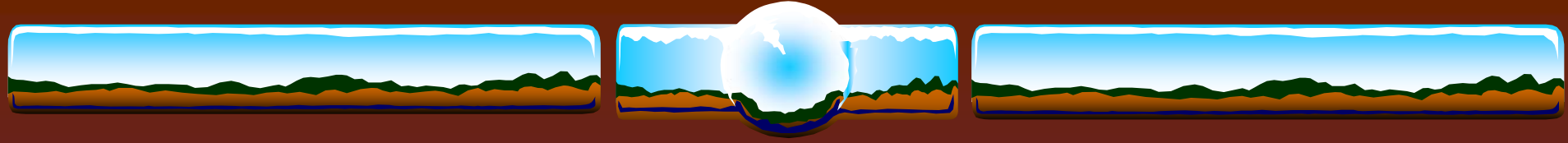
- SimH – Freeware
  - `simh.trailing-edge.com`
- Charon VAX – better, faster, stronger commercial product – faster than the fastest VAX hardware
  - `www.charon-vax.com`

```
Duo TTA0:> show system    ! At 37,000 feet
OpenVMS V7.3  on node DUO   2-APR-2005 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:>
```

Real & Simulated VAXen Performance



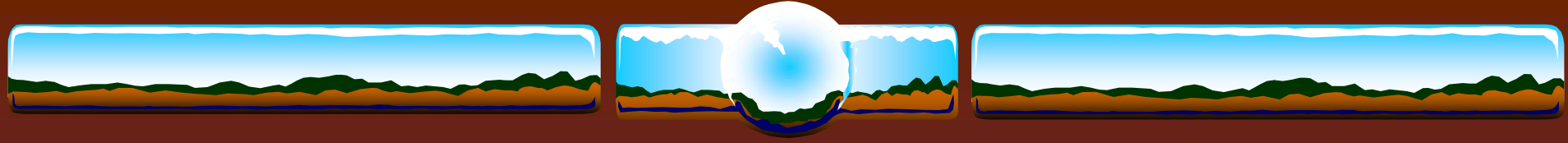
- Prime number generation
  - C program from Internet
  - Single-user
  - CPU intensive
- Charon-VAX
  - Intel Laptop 2ghz
  - …at 37,000 feet
- SimH machines
  - GS1280/1.15 32p
  - rx4640/1.5/6mb
  - Intel Laptop 2ghz



Tools & FreeWare

Don't Leave Home Without…

- GREP
- AWK
- TECO
- RZDISK
- ICALCV
- MBX
- ZIP & UNZIP
- DFU
- AlphaPatch (or VMS 8.2)
- RMS_TOOLS
- Ethereal
  `(http://www.ethereal.com/)`



QUESTIONS?

"Make your systems scream…
Not your users"

- anonymous…