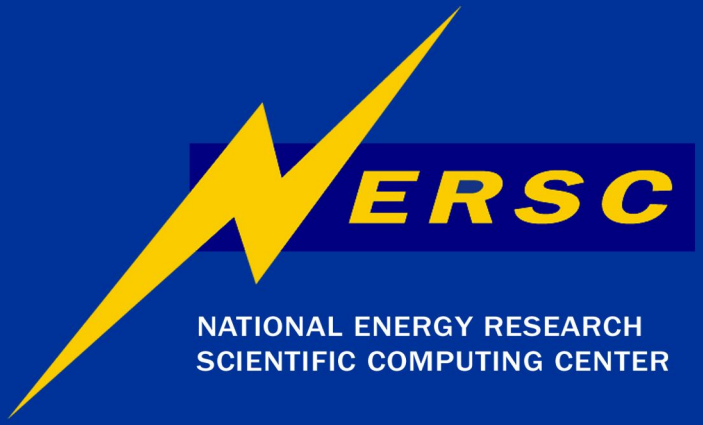


National Energy Research Scientific Computing Center (NERSC)

PSDF Activities

Cary Whitney (clwhitney@lbl.gov)
NERSC Center Division, LBNL
05/26/2004 2004 HEPiX

Outline

- CHOS
- ProcDN
- One-wire
- Capacitors
- Hot fixes
- St Michael/Patchfinder
- Linux virtual server
- Event monitoring
- SGE/LSF Utilization
- Lustre
- New projects

CHOS

- CHOS stands for chroot OS
- CHOS is a framework that simplifies running multiple Linux distributions *concurrently* on a single node
- This accomplished through a combination of the chroot system call, a loadable Linux kernel module, and a PAM module.
- It can also be integrated into the batch scheduler system and Globus
  - Modified job_starter to pick CHOS out of environment
- Runs under 2.4 and 2.6 kernels

CHOS System View

- RPM installs chos module, pam module, and creates framework directory (/chos).
- Copy or install alternate OS trees (ie. /auto/redhat8)
- Create /etc/chos.conf in OS tree (tells chos how to sanitize environment)
- Specify allowed OS trees in /etc/chos
- Run additional automounters (NFS mounts and local remounts in /chos area)

CHOS System Files

```
# cat /etc/chos
%SHELLS
/auto/redhat8:/auto/redhat8
rh73:/auto/redhat73
/auto/redhat73:/auto/redhat73
local:/local/root
rh62:/auto/common/os/redhat62
rh8:/auto/common/os/redhat8
rh9:/auto/common/os/redhat9
hepl30:/auto/common/os/hepl30
fc2:/auto/common/os/fc2

%ENV
```

```
# cat /etc/chos.conf
%ENV
CHOS
USER
LOGNAME
HOME
PATH
MAIL
SHELL
SSH_CLIENT
SSH_CONNECTION
SSH_TTY
TERM
DISPLAY
SSH_AUTH_SOCK
HOSTTYPE
VENDOR
OSTYPE
MACHTYPE
SHLVL
PWD
GROUP
HOST
REMOTEHOST
```

CHOS User View

- User creates .chos file that specifies preferred OS. Automatically switched to the OS on login

OR

- User sets CHOS to preferred OS and runs chos command to switch

PLUS

- Batch jobs automatically use OS that job was submitted under (currently works for SGE and LSF)

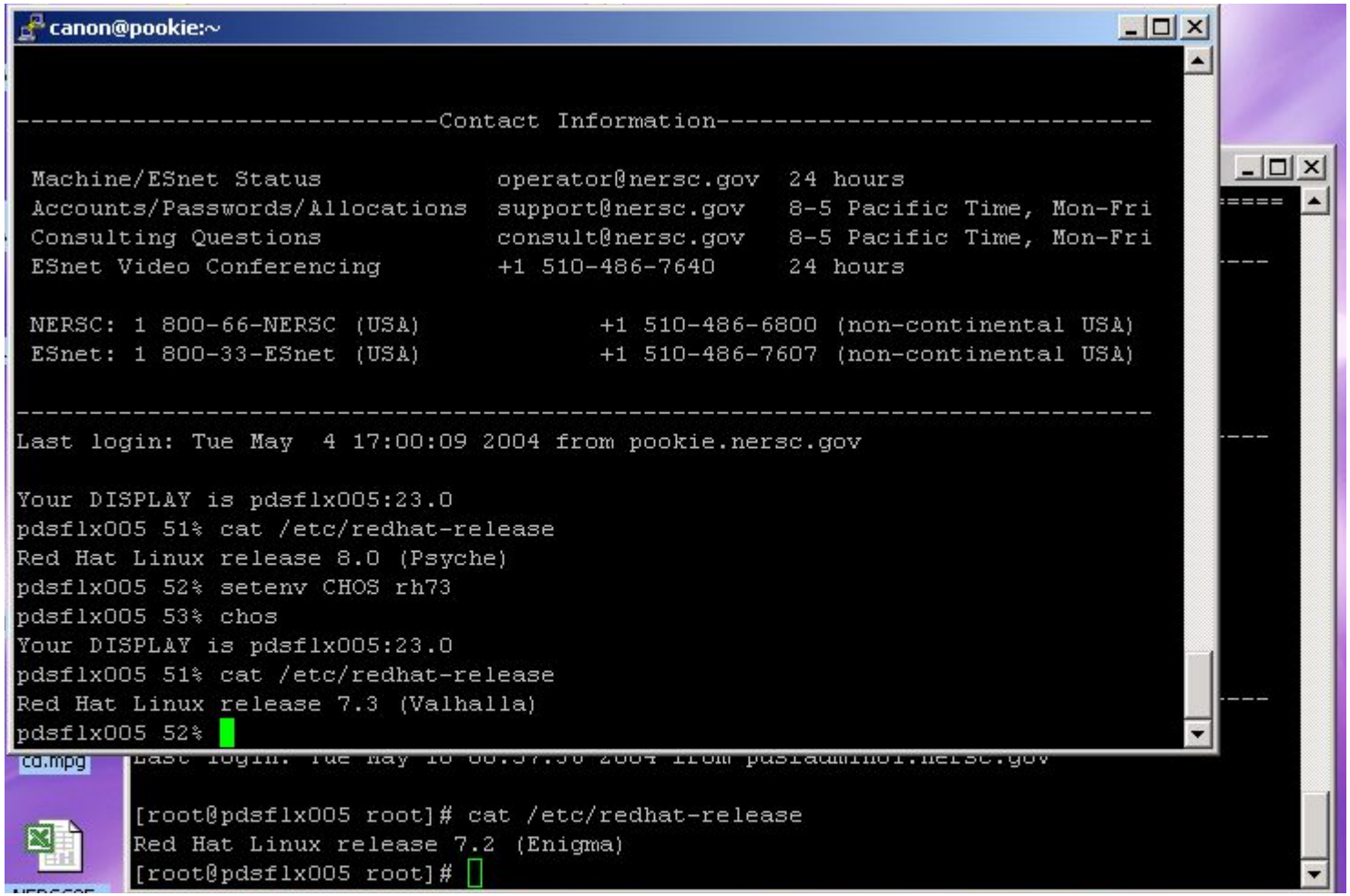
CHOS User File

$ cat .chos
/auto/redhat8

OR

$setenv CHOS rh73

CHOS In Action



CHOS Summary

- Could serve as a new model. For example…
  - A VO could distribute an entire OS tree that is maintained by the VO). The OS, applications, and environment would all be under the control of the VO. This shifts more responsibility to the VO.
  - Resource managers (sys admins) would be responsible for the kernel and services
  - CHOS could simplify Grid deployments in this scenario
- A paper is in the works
  Contact: canon@nersc.gov

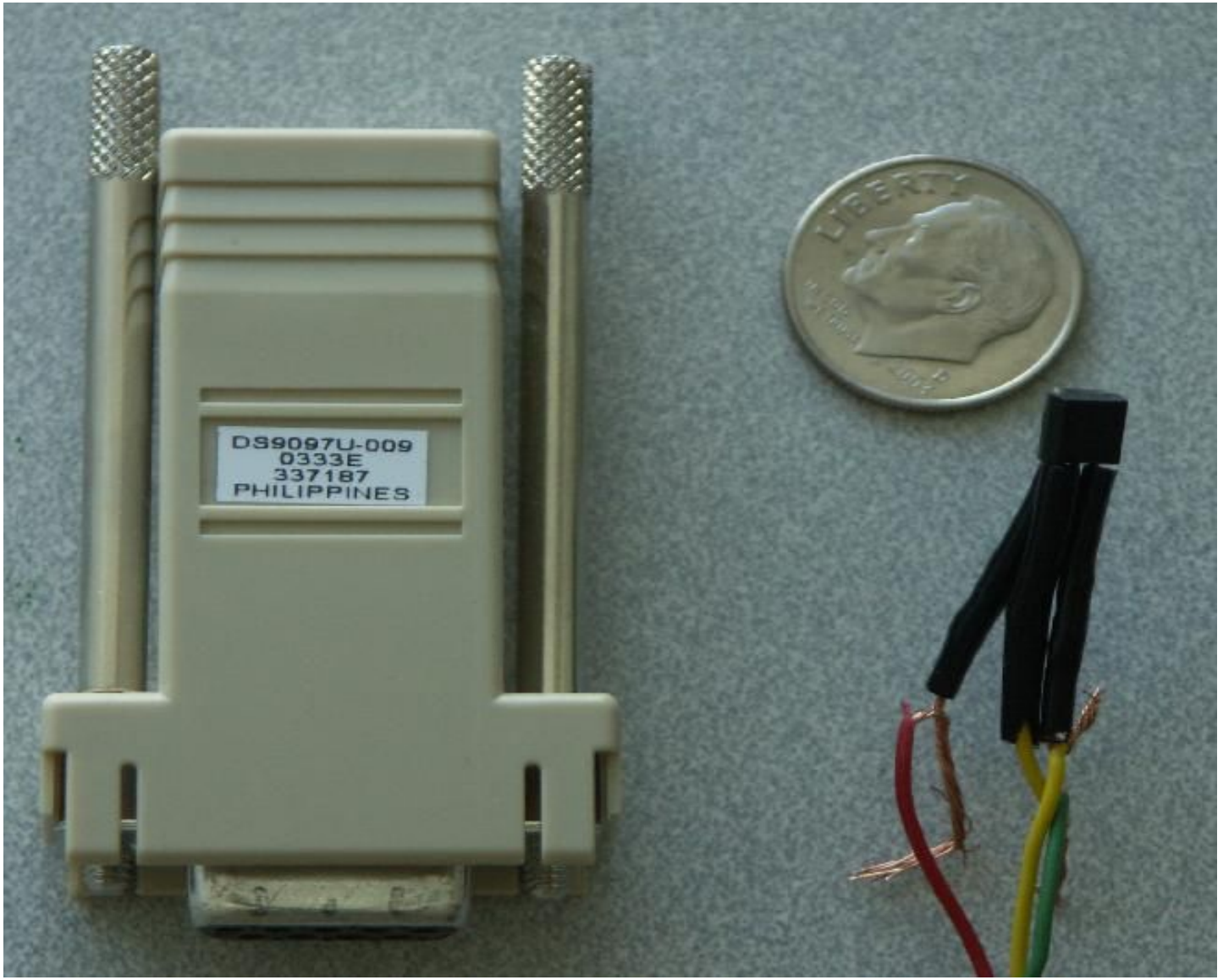
ProcDN

- kernel module maintains mapping between processes and a user's Globus distinguished name (DN)
- Modified gatekeepers (and other grid services) can initialize this mapping
- Modified gatekeeper for batch services also associates job id with the submitting DN (which is stored in a database)
- Modified job starters initializes the kernel mapping on the execution hosts (by querying DN from the database based on the job id)
  Contact: canon@nersc.gov

One-Wire

- Using One-wire serial interface. Connect one-wire network to cluster nodes to do:
  - Temperature sensing/rack profiling
  - Remote power management
  - Remote system reset
- Each node will have 1 device that can perform up to 8 functions (temp, power, reset, ...)
- 32 or so devices per rack all connected to a single serial port on the console server.

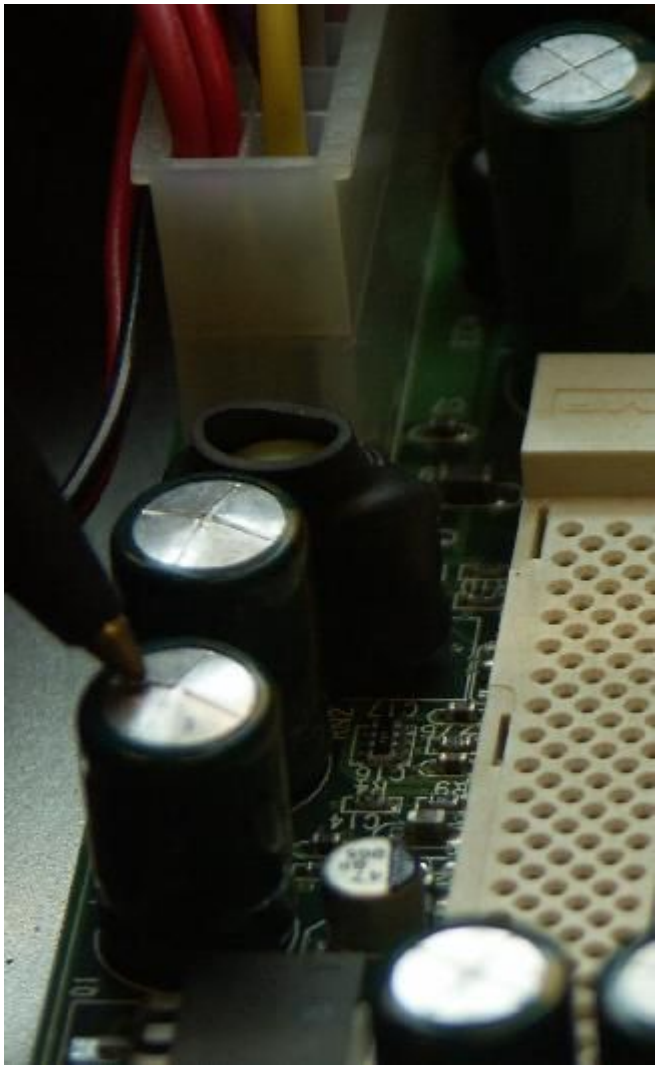
One-Wire Hardware



Office of Science
U.S. DEPARTMENT OF ENERGY

Capacitors

- Sporadic reboots/lockups
- Loss of a cpu
- Cause heat, stress and time
- Bad Capacitors
  - Rounded tops
  - Cracked tops
  - Bottoms pushed out
- 4 or 8 capacitors per board
  $.50 per and 15 minutes
- ~100 systems reclaimed
  Contact: tmlangley@lbl.gov



Office of Science
U.S. DEPARTMENT OF ENERGY

Bad Caps

Hot Fixes

- This is a direct benefit from the kernel class that we put on last summer.

  Fixes:
  - ptrace
  - mremap 1 & 2
  - brk

Hot Fix ptrace

```
int init_module(void)
{
  void **sys_call_table;

      lock_kernel();

      EXPORT_NO_SYMBOLS;
      printk("ptr1=0x%lx\n",(long)THIS_MODULE);
      printk("ptr2=0x%lx\n",&init_module);
      sys_call_table=find_sct();
      o_ptrace=sys_call_table[__NR_ptrace];
      sys_call_table[__NR_ptrace]=sys_call_table[31];
      unlock_kernel();
      return 0;
}
```

Hot Fixes mremap

```
asmlinkage unsigned long sys_mremap_wrapper(unsigned long addr,
        unsigned long old_len, unsigned long new_len,
        unsigned long flags, unsigned long new_addr) {

        unsigned long pnew_len;

        pnew_len = PAGE_ALIGN(new_len);

        if ((new_addr+pnew_len) >= 0xc0000000){
                printk("Suspicious behaviour: mremap %d\n",current->pid);
                printk("Suspicious behaviour: mremap 0x%lx 0x%lx\n",pnew_len,new_addr);
                return -ENOMEM;
        } else{
                return o_mremap(addr, old_len, new_len, flags, new_addr);
        }
}
```

Hot Fix brk

```
int init_module(void) {
 ...
  ptr=(unsigned char *)(do_brk);          /* We are looking for calls/jumps to this function */
  newptr=(unsigned char *)(my_brk);       /* This is what we will change it to               */
  for (cptr=start;cptr<end;cptr++){       /* Lets scan all of kernel space                   */
    if (*cptr==0xe8||*cptr==0xe9){        /* Look for calls or jumps                         */
      cptr++;                             /* If you find one look at the next dword          */
      lptr=(long *)cptr;
      cptr+=4;
      if ((cptr+*lptr)==(ptr)){           /* See if the offset would point to do_brk         */
        printk("fixing 0x%08lx\n",lptr);  /* If so, change it to our new routine             */
        *lptr=(newptr-cptr);
        count++;
       }
    } else{
      lptr=(long *)cptr;                  /* Look for the address as well.  This would show  */
      if ((unsigned char*)(*lptr)==ptr){  /* up in the symbol table.                         */
        printk("Fixing address at 0x%08lx\n",lptr);
        *lptr=(long)(newptr);
      }
    }
  }
  printk ("Fix brk installed..\n");       /* All done. */
  MOD_INC_USE_COUNT;                      /* We can't unload this one.  So lets inc the mod  */
                                          /* count and leave it there. */
  return 0;                               /* success */
}
```

Hot Fix brk cont

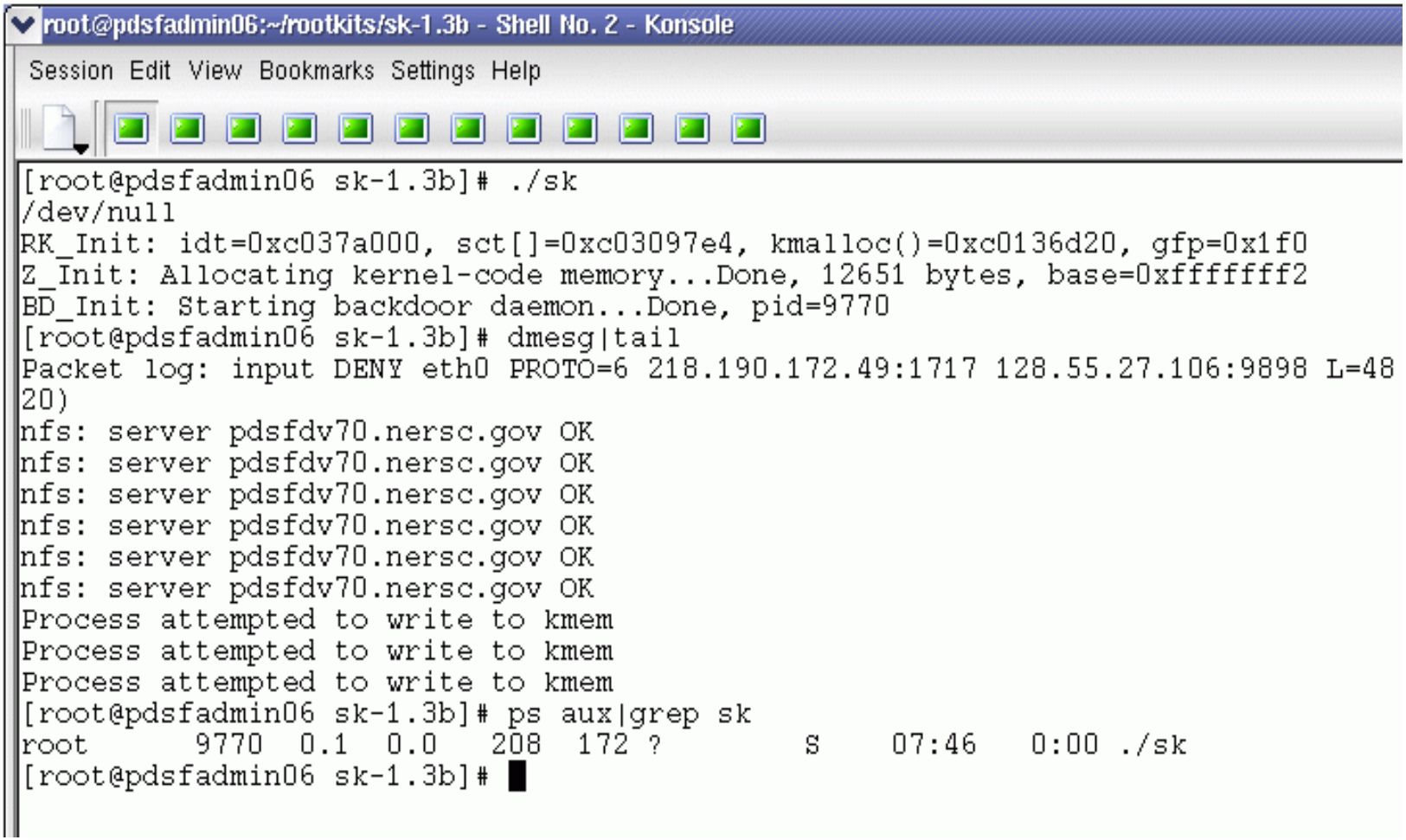
```
unsigned long my_brk(unsigned long addr, unsigned long len)
{
        len = PAGE_ALIGN(len);
        if (!len)
                return addr;

        if ((addr + len) > TASK_SIZE || (addr + len) < addr){    /* Let's make sure its in bounds */
                printk("caught do_brk exploit!!!\n");
                printk("pid: %d  uid:%d\n",current->pid,current->uid);
                return -EINVAL;
        }
        return do_brk(addr,len);                        /* Call the real do_brk         */
}
```

St Michael/Patchfinder

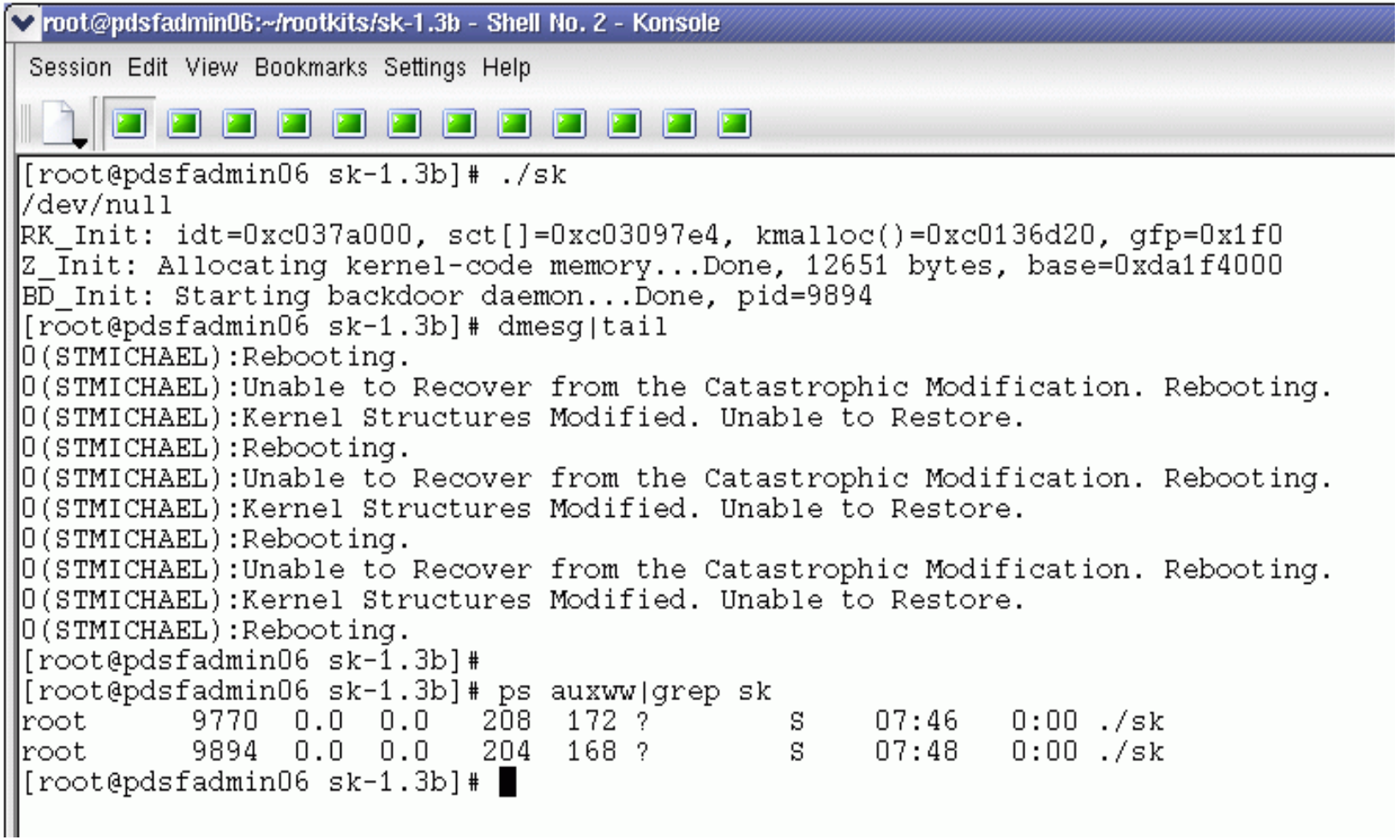
- St Michael
  - Kernel level integrity checker (finds changes caused by rootkits)
- Patchfinder
  - In kernel instruction counting, compares with known good system (search for rootkits)

Patchfinder Output

St Michael SucKit



St Michael SucKit 2



Linux Virtual Server

- Setup of a director with several mysql servers.
- Special module 'noarp' to keep real servers from responding to certain arp requests.

```
#setup:
#This script installs the VIP.
#The CIP, DIP and RIPs must be already installed,
#machines must be connected and be able to ping each other.
#CIP, RIPs usually are primary IPs on an interface.
#VIP, DIP are secondary (alias) IPs.
#
#          _________
#         | client  |
#         |_________|
#             | CIP=eth0 192.168.1.254
#             |      ____________
#             |------| director   |
#             |      |____________|
#             |  VIP=eth0:110 192.168.1.110/32
#             |  DIP=eth0:9   192.168.1.9
#             |
#       ------------------------
#       |                      |
# ________________    ________________
# | realserver1  |    | realserver2  |
# |______________|    |______________|
#   RIP1=eth0           RIP2=eth0
#  192.168.1.11         192.168.1.12
#
#        all realservers
#     VIP=lo:110=192.168.1.110 #
```

LVS Config

LVSCONF_FORMAT=1.1
LVS_TYPE=VS_DR
INITIAL_STATE=on
CLEAR_IPVS_TABLES=yes
VIP=eth0:110 pdsfdb00 255.255.255.255 pdsfdb00
#DIP line format - device[:alias] IP network netmask broadcast
DIP=eth0 pdsfcore03 128.55.24.0 255.255.252.0 128.55.27.255
SERVICE=t mysql wrr pdsfdb01 pdsfdb04 pdsfdb06
#SERVICE=t ftp rr RS1,1 RS2,2
#SERVICE=t http rr RS1 RS2
SERVER_VIP_DEVICE=lo:110
SERVER_NET_DEVICE=eth1
SERVER_GW=128.55.24.1

LVS Network

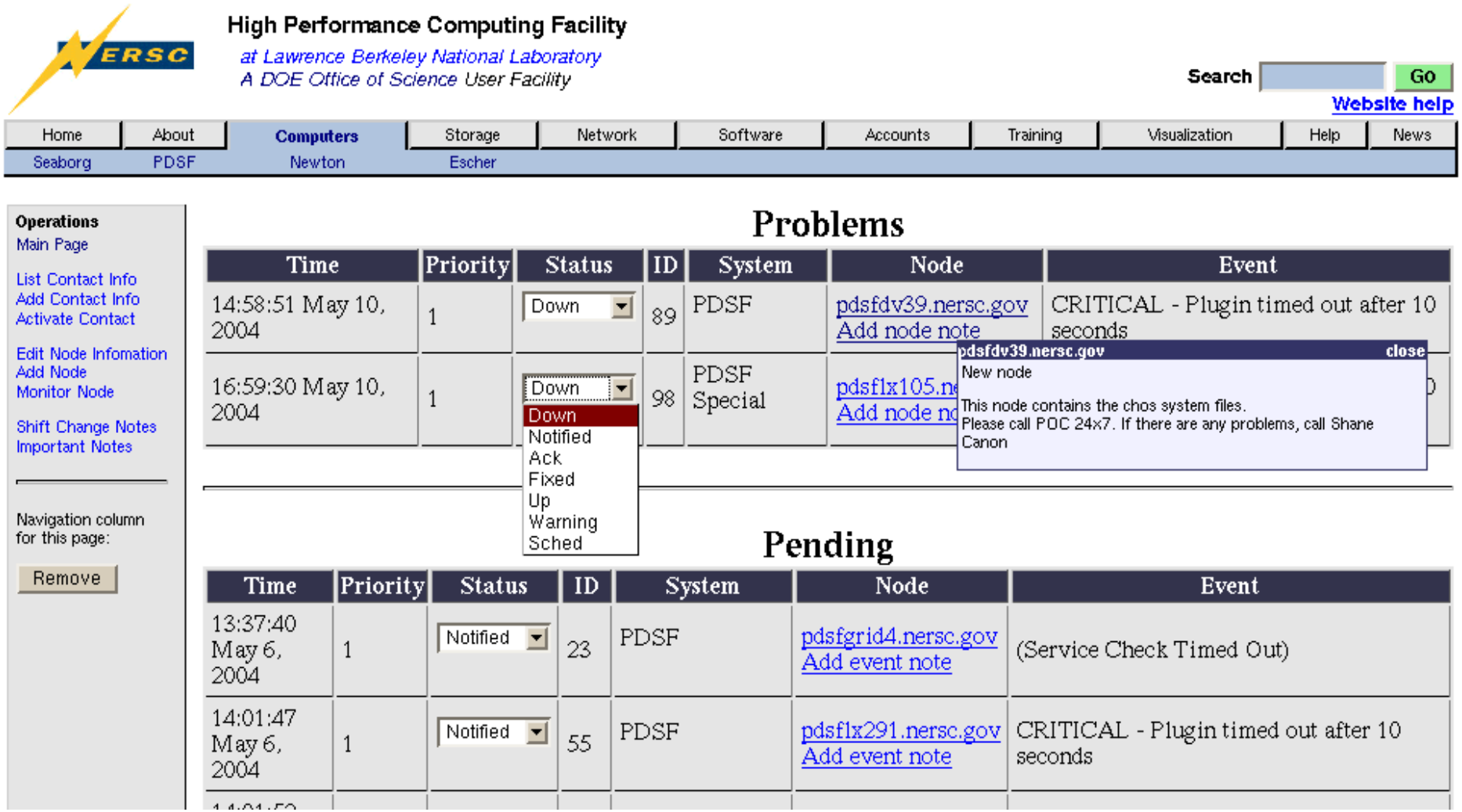
On Director:

```
eth0      Link encap:Ethernet  HWaddr 00:30:48:70:62:7F
          inet addr:128.55.24.17  Bcast:128.55.27.255  Mask:255.255.252.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11850792 errors:0 dropped:0 overruns:0 frame:0
          TX packets:403069 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:880354044 (839.5 Mb)  TX bytes:79219443 (75.5 Mb)
          Base address:0xc800 Memory:fe8e0000-fe900000

eth0:110  Link encap:Ethernet  HWaddr 00:30:48:70:62:7F
          inet addr:128.55.27.10  Bcast:128.55.27.10  Mask:255.255.255.255
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Base address:0xc800 Memory:fe8e0000-fe900000
```

LVS Admin

```
# ipvsadm
IP Virtual Server version 1.0.11 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  pdsfdb00.nersc.gov:mysql wrr
  -> pdsfdb06.nersc.gov:mysql     Route   1      0          0
  -> pdsfdb04.nersc.gov:mysql     Route   1      0          0
  -> pdsfdb01.nersc.gov:mysql     Route   1      0          0
# ipvsadm
IP Virtual Server version 1.0.11 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  pdsfdb00.nersc.gov:mysql wrr
  -> pdsfdb06.nersc.gov:mysql     Route   1      0          0
  -> pdsfdb04.nersc.gov:mysql     Route   1      0          1
  -> pdsfdb01.nersc.gov:mysql     Route   1      1          0
# ipvsadm
IP Virtual Server version 1.0.11 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  pdsfdb00.nersc.gov:mysql wrr
  -> pdsfdb06.nersc.gov:mysql     Route   1      1          0
  -> pdsfdb04.nersc.gov:mysql     Route   1      0          1
  -> pdsfdb01.nersc.gov:mysql     Route   1      0          1
```

Event Monitoring



Office of Science
U.S. DEPARTMENT OF ENERGY

Node Events

High Performance Computing Facility

at Lawrence Berkeley National Laboratory
A DOE Office of Science User Facility

Search Go

Website help

Home | About | **Computers** | Storage | Network | Software | Accounts | Training | Visualization | Help | News

Seaborg | PDSF | Newton | Escher

Operations

Main Page

List Contact Info
Add Contact Info
Activate Contact

Edit Node Infomation
Add Node
Monitor Node

Shift Change Notes
Important Notes

List of Events

Time	Node	Status	Weight	Event
17:50:30 Apr 27, 2004	pdsfgrid4.nersc.gov	Down	1	(Service Check Timed Out)
18:55:31 Apr 27, 2004	pdsfgrid4.nersc.gov	Fixed	1	Gatekeeper: Okay
10:52:33 Apr 28, 2004	pdsfgrid4.nersc.gov	Up	1	NULL
11:04:16 May 4, 2004	pdsfgrid4.nersc.gov	Fixed	1	Gatekeeper: Okay
11:46:01 May 4, 2004	pdsfgrid4.nersc.gov	Ack	1	NULL
15:05:21 May 4, 2004	pdsfgrid4.nersc.gov	Down	1	(Service Check Timed Out)
15:10:50 May 4, 2004	pdsfgrid4.nersc.gov	Fixed	1	Gatekeeper: Okay
21:11:21 May 5, 2004	pdsfgrid4.nersc.gov	Down	1	(Service Check Timed Out)
22:16:40 May 5, 2004	pdsfgrid4.nersc.gov	Fixed	1	Gatekeeper: Okay
13:37:40 May 6, 2004	pdsfgrid4.nersc.gov	Notified	1	NULL
13:38:37 May 6, 2004	See online documentation under the MISCELLANEOUS section for procedural instructions.			

Office of Science
U.S. DEPARTMENT OF ENERGY

Event Status

States Totals

Count	Status
57	Fixed
9	Ack
8	Notified
2	Warning
3	Down
1	Sched

POC on call

POC	System	Pager	On Coming POC
Shane Canon	CSG	[redacted]	Cary Whitney Go
Shane Canon	PDSF Special	[redacted]	Cary Whitney Go

Events

SGE vs LSF

- April
  - LSF – 153185 jobs on 396 processors
  - SGE – 53568 jobs on 180 processors
- YTD
  - LSF – 974095 jobs
  - SGE – 224518 jobs
- Support for parallel jobs, large number of jobs, grid, multiple groups, fair share scheduling, resource management, robust
- SGE Enterprise edition used – Source available

Lustre

- Looks promising for PDSF
- Performance (version 1.0.2):
  - Agg read: 252.3 MB/s
  - Agg write: 103.4 MB/s
- 8 lustre OST (Single 850 Mhz CPU/512 MB)
- 7 clients (Dual 2.6Ghz Xeon/2 GB)
- All GigE connected
- Quad bonded GigE connection between switches

Lustre Continue

- Problems
  - Poor recoverability from hardware failure (should be better in 1.2.1)
  - Configuration – Everyone needs to know about everyone else
  - Network timeouts could be better
  - Support? Model and pricing need work
- Contact: clwhitney@lbl.gov

Linux Auditing/Accounting

- Combine/Add comprehensive accounting and auditing to linux
  - CAS, CKRM, Enhanced Linux System Accounting (ELSA), perfctr, PAPI, systrace, light weight auditing, Secure Auditing for Linux (SAL)
- Currently surveying existing packages
- Taking requirements from Security, User Services, etc groups
- Inspiration from old Cray accounting and AIX POE++
- Goal: Late 2004/Early 2005 implementation
- Contact: canon@nersc.gov

Upcoming Projects

- SELinux via Fedora Core 2
- 10 GigE uplink
- Jumbo Frames
  - NFS Network
  - Connection to HPSS
- Filesystem tests (with GUPFS project)
  - GUPFS
  - ADICS

Conclusion

- A lot of activity. Contacts:
  - clwhitney@lbl.gov
  - canon@nersc.gov
  - tmlangley@lbl.gov
- Software available:
  - Real soon. Being finalized by Tech transfer department.