From AIX 7.2 announcement

IBM is introducing AIX 7.2, the IBM strategic UNIX operating system for mission-critical, core business applications, with the following features:

1. AIX Live Update for Interim Fixes.
2. Cluster Aware AIX (CAA) automation with repository replacement mechanism.
3. SRIOV-backed Virtual Network Interface Card (vNIC).
4. RDSv3 over RoCE, which adds support of the Oracle RDSv3 protocol over the Mellanox Connect RoCE adapters.
5. Flash Caching. Workloads can take advantage of a read-only cache.
6. DSO becomes part of AIX 7.2 (was a option extra at a cost).

ZP15-0527, dated October 5, 2015

AIX 7.2 Pre-Reqs

- POWER7, Power7+ or POWER8 or higher
  - No support for POWER5 or POWER6 or older

AIX 7.2 arrived on 4th December 2015

1 AIX 7.2 Live Kernel Update for Interim Fixes

Chris Gibson
- Power Systems Client Technical Specialist
- Melbourne, Australia,

- Excellent Web Article / whitepaper
“Holy grail” of OS upgrades is zero downtime
   – Various improvements for dynamic changes helped
   – But still non-trivial kernel changes need a reboot

Current AIX 7.2 uninterrupted update for Interim Fixes

Future AIX 7.2 uninterrupted update for SP & TL
   – Earlier than I initially expected

So what is the trick to get this technology miracle?
So what is the trick to get this technology miracle?

Workload Partitions

- 2007 AIX 6.1 was released
- Workload Partitions was a major feature (WPAR) with RBAC for WPAR security

- All familiar with Live Partition Mobility (LPM)
  - Jumps a whole AIX + Apps to a different server
- WPAR has Live Application Mobility (LAM)
  - Jumps just the Apps to a different AIX image

- Now add
  - Clone the source AIX and add the iFix
  - The two AIX LPARs are on the same machine
  - LAM the running Apps between Source and Target AIX
1 AIX 7.2 Live Kernel Update for Interim Fixes

Add disk(s) Alternative disk install fix

Remove Original Stop Original

Break off the disk & start in new LPAR

LAM

Fixed

Fixed

Fixed

Requirements:

- PowerVM and pure virtual AIX LPAR
- AIX 7.2
- HMC 840
- VIOS 2.2.3.5+
- FW 810 or later
- Two spare disks
- 100MB in /var root filesystem

No physical

vOptical, USB or Console (vTERM/vtmenu)
Control File: /var/adm/ras/liveupdate/lvupdate.data

- Two pages of comments at the top on the syntax options

**general:**
```
mode = automated
kext_check = no
```

**disks:**
```
nhdisk = hdisk1  # boot for surrogate
mhdisk = hdisk2  # mirror for surrogate
# tohdisk =     # optional: non-rootvg paging space
# tshdisk =     # optional: non-rootvg paging space
```

**hmc:**
```
lpars_id = 42
management_console = hmc14
user = hscroot
```

```bash
# chfs -a size=512M /var
```

**Commands:**

Authenicate with HMC:
```
hmcauth -u hscroot -a hmc_name
```

Preview check for: `/tmp/dummy.150813.epkg.Z`
```
geninstall -k -p -d /tmp dummy.150813.epkg.Z
```

Upgrade
```
geninstall -k -d /tmp dummy.150813.epkg.Z
```

*Note: the space character*
AIX 7.2 LKU Experience in beta testing

1. File & commands – easy to make mistakes/inconsistency
   – Like hmcauth one HMC but the other HMC in the file
2. Mandatory minimum of two other disks
3. All disks need to be multipath
4. No Virtual optical attached
5. No VTERM console
6. Can’t run the command from a VTERM console
7. Detailed logging is very good to work out the errors
8. Try a LPM Validate to help spot odd things!
9. Original LPAR renamed with added “_lk0” & finally removed

---

# vi /var/adm/ras/liveupdate/lvupdate.data

# hmcauth -u hscroot -a hmc14
Enter HMC password:

# clear; geninstall -k -d / dummy.150813.aix.k.z
Validating live update input data.
Computing the estimated time for the live update operation:

<table>
<thead>
<tr>
<th>LPAR: vm91.aix00cc.uk.ibm.com</th>
<th>Mode: F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blackout time(s): 69</td>
<td>Global_time(s): 525</td>
</tr>
</tbody>
</table>
Checking mirror vg device size:

```
Required device size: 5376 MB
Given device size: 3276 MB
PASSED: device size is sufficient.
Checking new root vg device size:
```
```
Required device size: 5376 MB
Given device size: 3276 MB
PASSED: device size is sufficient.
Checking temporary paging space device size:
```
```
Required device size: 512 MB
Checking temporary dump device size:
```
```
Required device size: 100 MB
Validating the adapters and their paths:
```
```
PASSED: adapters can be divided into two sets so that each has paths to all disks.
Checking other requirements:
```

---

general:
mode = automated
next_check = no
or preview
disks:
    nhdisk = hdisk1
    mhdisk = hdisk2
new rootvg
new rootvg mirror
hmc:
lpar_id = 42
management_console = hmc14
user = hscroot
```
PASSED: sufficient space available in /var.
PASSED: sufficient space available in /.
PASSED: no existing altinst_rootvg.
PASSED: rootvg is not part of a snapshot.
PASSED: pkcs11 is not installed.
PASSED: rootvg is not part of a snapshot.
PASSED: The trustchk Trusted Execution Policy is not on.
PASSED: The trustchk Trusted Library Policy is not on.
PASSED: The trustchk TSD_FILES_LOCK policy is not on.
PASSED: the boot disk is set to the current rootvg.
PASSED: the mirrorvg name is available.
PASSED: the rootvg is uniformly mirrored.
PASSED: the rootvg does not have the maximum number of mirror copies.
PASSED: the rootvg does not have stale logical volumes.
PASSED: all of the mounted file systems are of a supported type.
PASSED: this AIX instance is not diskless.
PASSED: no Kerberos configured for NFS mounts.
PASSED: multibios environment not present.
PASSED: Trusted Computing Base not defined.
PASSED: no local tape devices found.
PASSED: live update not executed from console.
PASSED: the execution environment is valid.
PASSED: enough available space for /var to dump Component Trace buffers.
PASSED: enough available space for /var to dump Light weight memory Trace buffers.
PASSED: all devices are virtual devices.
PASSED: No active workload partition found.
PASSED: nfs configuration supported.
PASSED: HMC token is present.
PASSED: HMC token is valid.
PASSED: HMC requests successful.
PASSED: Provided LPAR ID is available.
PASSED: A virtual slot is available.
PASSED: RSCT daemons are active.

PASSED: no Kerberos configuration.
PASSED: ipar is not remote restart capable.
PASSED: no virtual log device configured.
PASSED: ipar is using dedicated memory.
PASSED: the disk configuration is supported.
PASSED: no Generic Routing Encapsulation (GRE) tunnel configured.
PASSED: Firmware level is supported.
PASSED: vNIC resources available.
PASSED: Consolidated system trace buffers size is within the limit of 64 MB.
INFO: Any system dumps present in the current dump logical volumes will not be available after live update is complete.

Non-interruptable live update operation begins in 10 seconds.
Non-interruptable live update operation begins in 10 seconds.
Live AIX update in progress.

Initializing live update on original LPAR.
Validating original LPAR environment.
Beginning live update operation on original LPAR.
Requesting resources required for live update.

Notifying applications of impending live update.

Creating rootvg for boot of surrogate.

Starting the surrogate LPAR.

Creating mirror of original LPAR's rootvg.

Moving workload to surrogate LPAR.

Blackout Time started.

Blackout Time end.

Workload is running on surrogate LPAR.

Shutting down the Original LPAR.

The live update operation succeeded.

Live Application Mobility

Broadcast message from root@vm91.aixmcc.uk.ibm.com (pts/0) at 16:59:16 ...
Live AIX update completed.
File /etc/init.d has been modified.
One or more of the files listed in /etc/check_config files have changed.

# See /var/adm/ras/config.diff for details.

So how long do you think that takes for a small LPAR?
- 1 CPU
- 4GB RAM
- not much running

Minutes?

1 2 3 5 7 10 20 30
So how long do you think that takes for a small LPAR?
- 1 CPU
- 4GB RAM
- not much running

Minutes?
1  2  3  5  7  10  20  30

Notes:
- You never deal with WPAR directly for LKU
  - WPAR is default installed so no software added
  - WPAR build, used & removed in the background
- You do need CPU + RAM on the same machine to duplicate the LPAR
- You could LPM to a server with the spare resources, LKU and then LPM back
- LPM Verify is a good test for LKU readiness but you do need duplicate paths to disks
1 AIX 7.2 Live Kernel Update

Conclusions
1. Kernel team been thinking about this a long time
   = it is not a trivial problem
   DMA, Interrupts, function vector tables, virtual
   memory, Kernel pages can be pages out, . . .

2. Quite complicated but we have the technology

3. Staged arrival

4. Down side: few upgrades with reboots before
   we get full non-disruptive kernel SP/TL updates!

2 AIX Repacking
Network apps
2 AIX Repacking Network apps

Network applications are many
- Some are very old & very bad
- Some are know massive security holes telnet & ftp

- Problem pre-AIX 7.2 = two large AIX packages
  i.e. install all or nothing (and not an option)
  - bos.net.tcp.client
  - bos.net.tcp.server

- Some customers delete / disable unneeded stuff
  - Security hardening = good
  - But can causes dependency + update issues = bad
  - Next Service Pack or Technology Level upgrade
    They all get installed again !!!

- The repackage let you permantly remove “crufty”
  - Like telnet and FTP AIX packages from their build

- Old and New packages . . .
2 AIX Repacking Network

AIX 7.1 TL4

bos.net.ipsec.keymgt
bos.net.ipsec.rte
bos.net.ncs
bos.net.nfs.client
bos.net.nis.client
bos.net.snapp
bos.net.tcp.adt
bos.net.tcp.client
bos.net.tcp.client_core
bos.net.tcp.dfdp
bos.net.tcp.dhcp
bos.net.tcp.ftpd
bos.net.tcp.gated
bos.net.tcp.imapd
bos.net.tcp.nisd
bos.net.tcp.pop3d
bos.net.tcp.rcmd
bos.net.tcp.sendmail
bos.net.tcp.smit
bos.net.tcp.telnet
bos.net.tcp.telnetd
bos.net.tcp.tftp
bos.net.tcp.traceroute
bos.net.tcp.x500
bos.net.udc
bos.net.ucpc

All the network commands are in these 2 packages

Note: these two still exist.
Shell packages (nothing inside)
Used to install other packages & backward compatibility
Remove these BEFORE other packages like ftp & telnet
Other AIX 7.2 Packaging News

2 AIX Repackaging Network apps

- AIX 7.2 by defaults installs 695 packages
  - Removing packages = faster install
  - Switched off graphics & “old” box support = device drivers
  and got down to 200 packages
  - AIX 7.2 DVD1 = 3.2 GB  [AIX 7.1 DVD1 = 4.3 GB]

- Sys. Admin can remove unwanted packages from:
  - NIM
  - mksysb installs
  - PowerVC clones

- Good news:
  ssh not a default install but is on the Installer menu
AIX 7.2 Code Removal and LPP Changes

- No support for POWER6, POWER5, or POWER4
- Remove “Trusted Computing Base” → Trusted Execution
- Additional Code Removals from AIX 7.2
  - NIS+
  - NDAF
  - IBM Virtual Shared Disk (rsct.vsd)
  - IBM Systems Director Components; pConsole * Running Man!
  - Selected old adapters
  - Selected performance toolbox components & eclipse2.rte, including bos.perf.gtools and performance workbench GUI
  - IP over FC driver
  - Fcparray head driver
  - graPhigs
  - Java 5
  - Bos.lNed * Worst editor on UNIX
  - Obsolete locales
AIX 7.2 – Additional changes & enhancements

- **New**
  - OpenSSH is being added to the AIX Install menus
  - HTTPD support in NIM

- **Other changes**
  - CIFS Client – move it to the AIX Expansion Pack and provide the CIFS client with "as-is" support only.
  - JFS “Classic”: Remove as an install option; function would remain in AIX 7.2 and continue to be supported
  - DSO features in base AIX 7.2 OS (bos.aso) – not a separate LPP

- **LPPs not supported on AIX 7.2**
  - PowerSC Trusted Surveyor on AIX 7.2 as management server
  - Fast Connect
  - Performance Toolbox

3
POWER Flash Cache
think SSD

“cache_mgt” manual page states:
Manages the infrastructure that provides caching on the solid-state drive (SSD) devices
3 AIX SSD Cache

- Marketing was calling it POWER Flash Cache or now Flash Caching but it is **AIX only** (not IBM i or Linux)
- Marketing now Flash Caching using the term “flash” vaguely
- Here the “flash” means
  - internal SSD as a disk [Solid State Drive] or
  - internal SSD on an adapter
- Also not a USB Flash drive (memory key/pen drive)

3 POWER Flash Cache

Typical use
- Disk I/O read & write from FC disks as normal
- Always writes updated blocks to regular FC disk
- AIX caches a read copy of recently used blocks on faster local SSD device
- Next read satisfied from the read cache on SSD

Result:
- Higher performance - reduced read time
- Reduced SAN traffic
- Does not block LPM as “master” copy on FC disks
3 POWER Flash Cache

- Details
  - Workloads can be using physical storage or storage provisioned through FC, VIOS+vSCSI or VIOS+NPIV
  - Cache devices can be attached directly or provisioned through VIOS (vSCSI)
  - User may target individual or group of disks to be cached on AIX 7.2
  - Partition using a cache may use LPM with or without a locally attached flash

- Benefits
  - Most applications - higher throughput & lower latency
  - Completely invisible to applications

Non-default package
If cache_mgt “not found” then install it from AIX DVD media
Also brings in cache.mgt.rte package
No reboot needed

### Installation Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Level</th>
<th>Part</th>
<th>Event</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>cache.mgt.rte</td>
<td>7.2.0.0</td>
<td>USR</td>
<td>APPLY</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>cache.mgt.rte</td>
<td>7.2.0.0</td>
<td>ROOT</td>
<td>APPLY</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>bos.freed.rte</td>
<td>7.2.0.0</td>
<td>USR</td>
<td>APPLY</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>bos.freed.rte</td>
<td>7.2.0.0</td>
<td>ROOT</td>
<td>APPLY</td>
<td>SUCCESS</td>
</tr>
</tbody>
</table>

File /etc/inittab has been modified.
One or more of the files listed in /etc/check_config.files have changed.
See /var/adm/log/config.diff for details.

### Manual page for cache_mgt

- Google: cache_mgt
3 POWER Flash Cache with local drive

- **AIX**: Local
- **AIX**: Local
- **AIX**: Local

- **No cache**
- Simple private direct SSD for the cache
- Shared SSD via the VIOS for the cache

- **No LPM**
- **No LPM**
- **No LPM**

**Local internal disk slot Flash SSD or Flash on an adapter**

3 POWER Flash Cache with FC disks

- **AIX**: FC
- **AIX**: FC
- **AIX**: FC
- **AIX**: FC

- **No cache**
- **No cache**
- Simple private direct SSD for the cache
- Shared SSD via the VIOS for the cache

- **No LPM**
- **No LPM**
- **No LPM**
- LPM OK

**Local internal disk slot Flash SSD or Flash on an adapter**
3 POWER Flash Cache – cache architecture

**AIX cache is not a write via cache**

- AIX
- Cache
- Fast disks
- Slower disks

**It is a always written to disk with some blocks cached**

- AIX
- Cache
- Fast disks
- Some written
- Lots read

Result: the cache can be removed at any time with no issues as the FC disk(s) has 100% of the data

If a VIOS SSD for caching it can be removed to allow LPM

---

3 POWER Flash Cache with cached disks !!!

**Cache likely to be much faster**

- AIX deciding what gets
  - Flash cached
    - could be good
- Flash FC Disks used as cache
  - like FlashSystem V9000

**Cache may be faster**

- AIX cache with Disk cache
  - may not help
- Flash FC Disks used as cache

Nigel’s opinion:
Not prime target as may prove ineffective if both AIX SSD cache & FC disks have similar FC overhead & latency.
But note it does add further disk I/O bandwidth
QED: Benchmark recommended

---

© 2015 IBM
3 POWER Flash Cache - user selected hdisks

- **Cache Pool**: 1 or more hdisk
- **Cache Partition**: Disk slice from pool
- **Partition attach to hdisk or group of hdisks to cache**
- **Caching can be switched off and on**
- **You decide which hdisk or group of hdisks get caching**

**Supported “Flash” for cache use**

1. **Power SSD internal disk**
   - SSD in a Hard disk bay
   - Special SSD “credit card” via an internal SAS Controller

2. **PCIe2/PCIe3 SAS RAID adapter with write cache with SSD’s Attached**

3. **Power SSD disk in EXP24 External Disk Drawer**
For the future:

** cache_mgt cache list

** cache_mgt cache start {-t <targetDevName> -P <partitionName> | -t {<targetDevName> | all} | -P <partitionName> | [-f]}

** cache_mgt cache stop {-t {<targetDevName> | all} | -P <partitionName> | [all]} -p <partitionName> | [-f]}

** cache_mgt cache setup {-e [yes|no]} [-p <partitionName>] [-g <partitionName>]

** cache_mgt monitor start

** cache_mgt monitor stop

** cache_mgt monitor get {-h | -s | -h | -s}

Future:

** cache_mgt engine list [-t]

** cache_mgt engine register -n <cePath>

** cache_mgt engine unregister [-n <cePath>]

** cache_mgt partition create [-p <partitionName>] -s partitionSize [-P <partitionName>]

** cache_mgt partition unassign {-t <targetDevName> | [-P <partitionName>]}

** cache_mgt partition list [-l]

** cache_mgt cache start stop NOT AVAILABLE

** cache_mgt monitor NOT AVAILABLE

** cache_mgt mig get -r {-t {<targetDevName> | all} | [-P <partitionName>]} NOT AVAILABLE

** cache_mgt mig set -r [yes | no] {-t {<targetDevName> | all} | -P <partitionName>]

** cache_mgt pool extend [-p <poolName>] -d <devName> [,<devName>,...]

** cache_mgt pool extend [-p <poolName>] -d <devName> [,<devName>,...]

** cache_mgt pool remove [-p <poolName>] [-f]
3 Example of suitable disks for the cache

```
# lsdev | grep hdisk
hdisk0  Available Virtual SCSI Disk Drive ← my SSP
hdisk1  Available 01-00-00 SAS Disk Drive
hdisk2  Available 01-00-00 SAS 4K Solid State Drive
hdisk3  Available 01-00-00 SAS 4K Solid State Drive
hdisk4  Available 01-00-00 SAS 4K Solid State Drive
hdisk5  Available 01-00-00 SAS 4K Solid State Drive

# cache_mgt device list
hdisk2
hdisk3
hdisk4
hdisk5
#
```

3 POWER Flash Cache part 1 of 2 Setup

AIX Physical device mode set-up
- Create a cache pool from list of cache devices
  ```
  # cache_mgt pool create -d hdisk1 -p pool1
  Pool pool1 created with device hdisk1
  ```
- Create cache partition in the pool & list the partition
  ```
  # cache_mgt partition create -p pool1 -s 80M -P part1
  Partition part1 created in pool pool1.
  ```
- Assign partition to a target disk
  ```
  # cache_mgt partition assign -t hdisk2 -P part1
  Partition part1 assigned to target hdisk2.
  ```
- Start caching of a target device & list the state
  ```
  # cache_mgt cache start -t hdisk2
  Cache for target hdisk2 has been started.
  ```
3 Behind the covers

The cache **pool** is a LVM Volume Group

```bash
# lsdev | grep cache
```

The cache **partition** is a LVM Logical Volume

```bash
# lsvg -l pool1
```

Also device devices:

```bash
# lsdev | grep cache
```

- When the cache is started the LV state = **open/syncd**

```
Also device devices:

# lsdev | grep cache
cache0 Available SSD Cache virtual device
cengine0 Available SSD Cache engine
```

3 Behind the covers

Did you notice many command options are optional?

```bash
# cache_mgt pool create -d hdisk4 ➩ not optional
Pool cmpool0 created with devices hdisk4.
```

```bash
# cache_mgt partition create -s80M ➩ just size
Partition cmpart0 created in pool cmpool0.
```

```bash
# cache_mgt partition assign -t hdisk1 ➩ just target
Partition cmpart0 assigned to target hdisk1.
```
3 POWER Flash Cache part 2 of 2 Admin

AIX Physical device mode admin

- List the state
  ```
  # cache_mgt pool list
  pool1, hdisk1
  # cache_mgt partition list -l
  part1, pool1
  # cache_mgt cache list
  hdisk2, part1 active
  ```

- Grow the cache pool
  ```
  # cache_mgt pool extend -p pool1 -d hdisk5 -f
  Pool pool1 extended with device hdisk5.
  ```

- Extend an existing cache partition size
  ```
  # cache_mgt partition extend -P part1 -s 120M
  Partition part1 extended to size 120M.
  ```

3 The un-make commands

There is also the undo commands

```bash
# cache_mgt cache stop -t hdisk2
# cache_mgt cache stop -t all

# cache_mgt partition unassign -t hdisk2
# cache_mgt partition remove ...

# cache_mgt pool remove ...
```
**Warning in this first release**

```bash
# cache_mgt pool create -d hdisk3
Failed to create pool:
Maximum number of cache pools (1) exceeded.

# cache_mgt partition create -s80M
Failed to create partition:
Maximum number of cache partitions (1) exceeded.
```

```bash
cache_mgt command manual page:
Only a single cache pool is supported in the physical mode and
caching can be started only on a single cache partition.
```

The command syntax suggests later releases might allow
- Multiple cache pools &
- Multiple cache partitions

**POWER Flash Cache via a Virtual I/O Server**

- cache_mgt on a VIOS supports many LPARs
- Partly set up on the VIOS (as root = oem_setup_env)
  - Create pool
  - Create partition
  - Assign cache device to LPAR
- Partly on AIX LPAR(s)
  - Assign cache device to regular hdisk
  - Cache start
3 POWER Flash Cache via VIOS

AIX Physical device mode set-up **VIOS side**
- Create a cache pool from list of cache devices
  
  ```bash
  # cache_mgt pool create -d hdisk1 -p pool1
  Pool pool1 created with device hdisk1
  ```
- Create cache partition in the pool & list the partition
  
  ```bash
  # cache_mgt partition create -p pool1 -s 80M -P part1
  Partition part1 created in pool pool1.
  ```
- Assign partition to a target client LPAR
  
  ```bash
  # cache_mgt partition assign -v vhost1 -P part1
  Partition part1 assigned vSCSI host adapter vhost0.
  ```

AIX Physical device mode set-up **AIX LPAR side**
- List the cache devices
  
  ```bash
  # cfmgmgr ; lsdev | grep cachedisk
cachedisk0 Available Virtual SCSI Solid State Drive
  ```
- Assign cache device the target disk
  
  ```bash
  # cache_mgt partition assign -t hdisk2 -P cachedisk0
  Partition cachedisk0 assigned to target hdisk2.
  ```
- Start caching of a target device & list the state
  
  ```bash
  # cache_mgt cache start -t hdisk2
  Cache for target hdisk2 has been started.
  ```

---

**3 POWER Flash Cache via VIOS**

```
$ cache_mgt
rksh: cache_mgt: 0403-006 Execute permission denied.
$ oem_setup_env

# cache_mgt device list
  
  hdisk1
  hdisk2
  hdisk3
  hdisk39

# cache_mgt pool create -d hdisk1
  Pool cmpool0 created with devices hdisk1.

# cache_mgt pool create -d hdisk2
  Pool cmpool1 created with devices hdisk2.

# cache_mgt partition create -s 256G
  ⇔ missing -p option

Failed to create partition:
There is more than one pool hence the pool cannot be automatically selected.

# cache_mgt partition create -s 64G -p cmpool1
  ⇒ allow default partition name (-P)
  Partition cmpart0 created in pool cmpool0.

# cache_mgt partition create -s 64G -p cmpool1
  ⇒ allow default partition name (-P)
  Partition cmpart1 created in pool cmpool1.
```
### 3 POWER Flash Cache via VIOS

**Deliberate error: Not enough space in the pool**

```
# cache_mgt partition create -s 64G -p cmpool1
Failed to create partition cmpart2 in pool cmpool1: ← error are a LV create failures
Failed to execute command '/usr/sbin/mklv -y cmpart2 cmpool1 64G':
Return Code: 1
Standard Error:
Failed to execute command '/usr/sbin/mklv -y cmpart2 cmpool1 64G':
# cache_mgt partition create -s 64G -p cmpool1
```

```
# lsdev | grep cachedisk
```

```
Partition cachedisk0 assigned to target hdisk0.
```

```
Deliberate error: Not enough space in the pool to keep strictness and satisfy allocation requests. The command should be retried with different allocation characteristics.
```

```
0516-822 mklv: Unable to create logical volume.
```

### 3 POWER Flash Cache via VIOS on AIX

```
# cfgmgr
# cache_mgt device list ← hdisk(s) are on the VIOS not here in AIX
#
# lsdev | grep cachdisk
cachedisk0 Available Virtual SCSI Solid State Drive
```

```
# lsdev | grep cache
cache0 Defined SSD Cache virtual device ← device driver
cachedisk0 Available Virtual SCSI Solid State Drive ← actual cache
cengine0 Defined SSD Cache engine ← cache algorithm
```

```
# lsvp
hdisk0 00f9d4944a23de64 rootvg active
cachdisk0 none None
```

```
# cache_mgt partition assign -P cachdisk0 -t hdisk0
Partition cachdisk0 assigned to target hdisk0.
```

```
# cache_mgt cache start -t hdisk0
Cache for target hdisk0 has been started.
```
cache_mgt Cheat Sheet

- Test Config: HDD = hdisk6 and SSD = hdisk2
- cache_mgt device list → Output your online SSD’s suitable for caching
- cache_mgt pool create -d hdisk2 -p cmpool1
- cache_mgt pool list -l
- cache_mgt partition create -p cmpool1 -s 32G -P cmpart1
- cache_mgt partition list -l
- cache_mgt partition assign -t hdisk6 -P cmpart1
- cache_mgt cache start -t hdisk6
- Cache_mgt cache stop -t hdisk6
- cache_mgt cache list
- cache_mgt monitor get

Simplistic “Does it work? test

- Local HHD
- 100 GB on JFS2
- Direct I/O
- 4KB blocks
- 80% read + 20% write
- Random
- 8 processes doing I/O
- 8 x 1 GB file
- at 522 IOPS

- Switch on cache
- 32 GB
- At 1737 + 475 IOPS
- = 2212 IOPS

- 4.23 times faster
  - Have seen that the cache warms up over time – wait at least 5 minutes or longer
  - Writes to real disk
  - Reads from cache disk
cache_mgt monitor -s

# cache_mgt monitor get -s

ETS Device I/O Statistics -- hdisk6

Read Count: 788298
Write Count: 197124
Read Hit Count: 751278
Partial Read Hit Count: 0
Read Bytes Xfer: 3228686808
Write Bytes Xfer: 807419004
Read Hit Bytes Xfer: 3077234688
Partial Read Hit Bytes Xfer: 0
Promote Read Count: 14017363968
Promote Read Bytes Xfer: 13368

I need to study these more during a real workload or benchmark!
- Get in touch.

POWER Flash Cache Conclusions:

- Flexible design with cache pool & cache partitions
- SSD’s directly physical at AIX level or via VIOS
- Cache is Transparent (no application changes)
  - Always writes to the real disks as the master copy
- Cache target can be single disk or a disk group
- LPM possible
  - SSD Cache on the VIOS – just works
  - SSD Cache on AIX – remove SSD(s) before LPM
- The slower the normal disks, the bigger the effect

Notes:

- AIX LPAR min 4GB
- No “shared disks” for Workload data or for the cache
  - “Shared disk” meaning online to more then one VIOS or AIX
Power Systems Technical Webinars

AIX 7.2
- Live Kernel Update
- Network LPP rework
- POWER Flash Cache

Are you keeping up to date?

mr_nmon on twitter
- Only used to POWER / AIX technical content, hints, tips and links

YouTube™ 131 techie hands-on videos on YouTube at
http://www.youtube.com/nigelargriffiths

AIXpert Blog
- Lots of mini articles & thoughts
  http://tinyurl.com/AIXpert

Also:
- http://tinyurl.com/ibmAIXVUG
- http://tinyurl.com/PowerSystemsTechnicalWebinars
An aside on internal SSD disks on my E850

- Might save you some time!

---

- Using Internal SSD on a E850

```
Command: OK          stdout: yes          stderr: no
Before command completion, additional instructions may appear below.
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Resource</th>
<th>State</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>sissas0</td>
<td>FFFFFFFF</td>
<td>Secondary</td>
<td>PCIe3 x8 SAS RAID Internal Adapter 6Gb</td>
<td></td>
</tr>
<tr>
<td>tsosix</td>
<td>FFFFFFFF</td>
<td>HA Linked</td>
<td>Remote adapter SN 0068412E</td>
<td></td>
</tr>
<tr>
<td>hdisk11</td>
<td>FC0000FF</td>
<td>Optimal</td>
<td>RAID 0 Array</td>
<td>139.6GB</td>
</tr>
<tr>
<td>pdisk2</td>
<td>0000006FF</td>
<td>Active</td>
<td>Array Member</td>
<td>N/A</td>
</tr>
<tr>
<td>hdisk12</td>
<td>FC0100FF</td>
<td>Optimal</td>
<td>RAID 0 Array</td>
<td>139.6GB</td>
</tr>
<tr>
<td>pdisk3</td>
<td>000007FF</td>
<td>Active</td>
<td>Array Member</td>
<td>N/A</td>
</tr>
<tr>
<td>hdisk13</td>
<td>FC0200FF</td>
<td>Optimal</td>
<td>RAID 0 Array</td>
<td>139.6GB</td>
</tr>
<tr>
<td>pdisk1</td>
<td>000001FF</td>
<td>Active</td>
<td>Array Member</td>
<td>N/A</td>
</tr>
<tr>
<td>hdisk14</td>
<td>FC0300FF</td>
<td>Optimal</td>
<td>RAID 0 Array</td>
<td>139.6GB</td>
</tr>
<tr>
<td>pdisk0</td>
<td>000000FF</td>
<td>Active</td>
<td>Array Member</td>
<td>N/A</td>
</tr>
<tr>
<td>pdisk4</td>
<td>000408FF</td>
<td>Active</td>
<td>4K RI Array Candidate</td>
<td>N/A</td>
</tr>
<tr>
<td>pdisk5</td>
<td>000409FF</td>
<td>Active</td>
<td>4K RI Array Candidate</td>
<td>N/A</td>
</tr>
<tr>
<td>pdisk6</td>
<td>00040AFF</td>
<td>Active</td>
<td>4K RI Array Candidate</td>
<td>N/A</td>
</tr>
<tr>
<td>pdisk7</td>
<td>00040BFF</td>
<td>Active</td>
<td>4K RI Array Candidate</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Hard Disks in an array as members (in use)

SSD’s in RAID format + Array Candidate but can’t be added to an array
- Using Internal SSD on a E850

Command: OK  stdout: yes  stderr: no

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Resource</th>
<th>State</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>sissas1</td>
<td>FEFFFFF</td>
<td>Primary</td>
<td>PCIe3 x8 SAS RAID Internal Adapter 6Gb</td>
<td></td>
</tr>
<tr>
<td>sissas0</td>
<td>FEFFFFF</td>
<td>HA Linked</td>
<td>Remote adapter SN 0055T010</td>
<td></td>
</tr>
<tr>
<td>pdisk1</td>
<td>000001FF</td>
<td>Active</td>
<td>Array Candidate</td>
<td>139.6GB</td>
</tr>
<tr>
<td>pdisk2</td>
<td>000006FF</td>
<td>Active</td>
<td>Array Candidate</td>
<td>139.6GB</td>
</tr>
<tr>
<td>pdisk3</td>
<td>000007FF</td>
<td>Active</td>
<td>Array Candidate</td>
<td>139.6GB</td>
</tr>
<tr>
<td>hdisk6</td>
<td>000000FF</td>
<td>Available</td>
<td>SAS Disk Drive</td>
<td>146.8GB</td>
</tr>
<tr>
<td>hdisk2</td>
<td>00040FFF</td>
<td>Available</td>
<td>SAS 4K Solid State Dr</td>
<td>200.0GB</td>
</tr>
<tr>
<td>hdisk3</td>
<td>000409FF</td>
<td>Available</td>
<td>SAS 4K Solid State Dr</td>
<td>200.0GB</td>
</tr>
<tr>
<td>hdisk4</td>
<td>00040AFF</td>
<td>Available</td>
<td>SAS 4K Solid State Dr</td>
<td>200.0GB</td>
</tr>
<tr>
<td>hdisk5</td>
<td>000408FF</td>
<td>Available</td>
<td>SAS 4K Solid State Dr</td>
<td>200.0GB</td>
</tr>
</tbody>
</table>

---

IBM SAS Disk Array Manager

- Change/Show SAS pdisk Status
- Delete an Array Candidate pdisk and Format to JBOD block size

![Diagram of disk array configuration with labels indicating SSD in JBOD, hard disks in an array, and SSD in JBOD format & appears as hdisk ready for AIX cache use.]