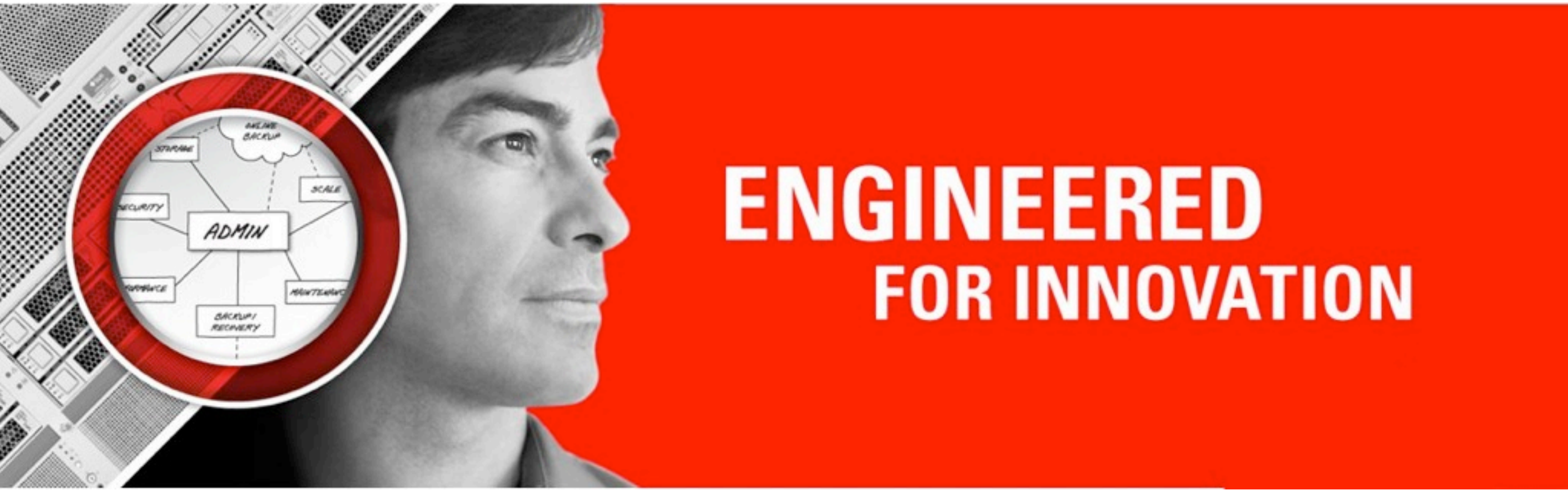

ORACLE

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®

Solaris 11: New Features in ZFS

Constantin Gonzalez
Oracle EMEA Engineered Systems Architecture Group

Oracle Solaris 11 ZFS New Features

- ZFS root/boot/install and patch/update features
- Deduplication
- Encryption
- Read-only pool import
- Pool import with missing log device
- `zpool status` and `zpool list` improvements
- `zfs send` and `zfs receive` backup options
- `zfs diff` to compare snapshot differences
- ACL interoperability improvements
- Performance improvements

ZFS Root/Boot/Install/Update

- ZFS is the default root file system. No UFS for root.
- Boot environments: See `beadm(1M)`
- New keywords in Auto Installer (AI) for swap and dump volume resizing
- IPS tightly integrated with boot environments
  - Uses ZFS snapshots and clones
- Mirrored root pool improvements
  - Automatic boot block application when attaching a disk to a root pool for mirroring.
  - But not when `autoreplace` property is used.

ZFS Deduplication

- Can be switched on/off at any time, off by default
- `zdb -S poolname` determines possible savings
- Synchronously: While data is written, block-based
- Dedup table tracks every block's checksum
- Uses 256-Bit checksums to find duplicates
  - SHA-256 recommended, extra verification optional
- 320 Byte RAM per block for optimal performance
  - Example: 20 TB @ 128K blocksize => ca. 50 GB of RAM
  - Example: 1 TB @ 64K blocksize => ca. 5 GB of RAM
  - Having SSDs für L2ARC can help
- Switch on with `zfs set dedup=on pool/fs`
- Works with compression and encryption

ORACLE

To Dedupe or not to Dedupe?

- Figure out what savings you can get:
  - Test with representative real data, then check the `DEDUP` column in the `zpool list` output.
  - Simulate dedupe davings by using `zdb -S`
  - Perform an educated guest: How much overlap do you expect?
- Figure out the price to pay:
  - Find total number of blocks. Either with `zdb -b`, or by estimating average blocksize, then dividing space/blocksize.
  - 320 * Number_of_blocks = Dedupe table size
  - Should fit into 1/4 or (RAM - 1 GB) for optimal performance
- Rules of thumb:
  - For each TB of pool data, expect 5 GB of dedupe table.
  - 20 GB of RAM per TB of pool data for optimum performnance
  - Using L2ARC is a good compromise

ORACLE

ZFS Encryption

- Per-dataset encryption, enabled at creation time
  - Inherited by snapshots/clones, can't be removed
- Key sources: prompt, passphrase, file
- `zfs create -o encryption=on tank/foo`
- A wrapping key encrypts the data encryption keys
  - Can be changed at any time
  - Default encryption algorithm: `aes-128-ccm`, can be changed
- Swap, `/var/tmp` can be encrypted as well
- Not bootable
- `zfs send/receive` are not encrypted

ZFS Encryption Examples

```
# zfs create -o encryption=on tank/home/darren
Enter passphrase for 'tank/home/darren': xxxxxxxx
Enter again: xxxxxxx

# pktool genkey keystore=file outkey=/dmkey.file keytype=aes
keylen=256

# zfs create -o encryption=aes-256-ccm -o
keysource=raw,file:///dmkey.file tank/home/darren

# zfs key -c tank/home/darren
Enter new passphrase for 'tank/home/darren': xxxxxxxx
Enter again: xxxxxxxx
```

Pool Import Recovery

- Pools can now be imported with a missing log
  - log devices can be reattached after import
  - Run `zpool clear` to clear any errors
- Automatic recovery of broken pools
  - Traverses transaction history until it finds a valid TXG
- Read-Only Pool Import:

```
# zpool import -o readonly=on tank
# zpool scrub tank
cannot scrub tank: pool is read-only
```

ZFS Send/Receive Enhancements

- Change file system property values while send/ receiving ZFS snapshot stream:

```
# zfs get compression tank/data
NAME        PROPERTY     VALUE     SOURCE
tank/data  compression  off       default
# zfs send -p tank/data@snap1 |
  zfs recv -o compression=on -d bpool
# zfs get -o all compression bpool/data
NAME        PROPERTY     VALUE  RECEIVED  SOURCE
bpool/data  compression  on     off       local
```

ZFS Send/Receive Enhancements

- Supports “inheriting” properties from received stream:

```
# zfs send -b bpool/data@snap1 | zfs recv -d restorepool
# zfs get -o all compression restorepool/data
NAME                PROPERTY     VALUE     RECEIVED  SOURCE
restorepool/data  compression  off       off       received
```

ZFS Send/Receive Enhancements

- Properties can be disables at receive time:

```
# zfs send -R tank/home@1020 | zfs recv -x quota bpool/home
# zfs get -r quota bpool/home
NAME                        PROPERTY  VALUE  SOURCE
bpool/home                  quota     none   default
bpool/home@1020             quota     -      -
bpool/home/cindys           quota     none   local
bpool/home/cindys@1020      quota     -      -
bpool/home/tom              quota     none   local
bpool/home/tom@1020         quota     -      -
```

ZFS Snapshot Differences

```
$ ls /tank/home/timh
fileA
$ zfs snapshot tank/home/timh@old
$ ls /tank/home/timh
fileA fileB
$ zfs snapshot tank/home/timh@new
$ zfs diff tank/home/timh@old tank/home/timh@new
M       /tank/home/timh/
+       /tank/home/timh/fileB
```

Legend: `M` = Modified, `-` = Removed, `+` = Added, `R` = Renamed

Miscellaneous New Features

- `rstchown` per dataset instead of `/etc/system`
- ACL interoperability improvements
- `logbias=latecy | throughput`
- `sync=standard|always|disable`
- `primarycache=none|all|metadata`
- `secondarycache=none|all|metadata`
- `mlslabel` – for Trusted Extensions zones

ZFS Performance Improvements

- Better, faster block allocator
- `zpool scrub` now uses prefetch
- Raw scrub/resilver
- Zero-copy I/O for NFS&CIFS
- Explicit sync mode control (`sync` property)
- RAID-Z/mirror hybrid allocation

More Information

- Solaris 11 “What’s New” Documentation
- Encryption: Darren Moffat’s Weblog
  - http://blogs.oracle.com/darren
  - (Thanks, Darren for slides and examples!)
- ZFS Community on OpenSolaris.org
  - Including the ZFS Dedup FAQ
- Deduplication: “Constant Thinking” Blog
  - http://constantin.glez.de/

Q&A

Hardware and Software

ORACLE

Engineered to Work Together

ORACLE


ORACLE