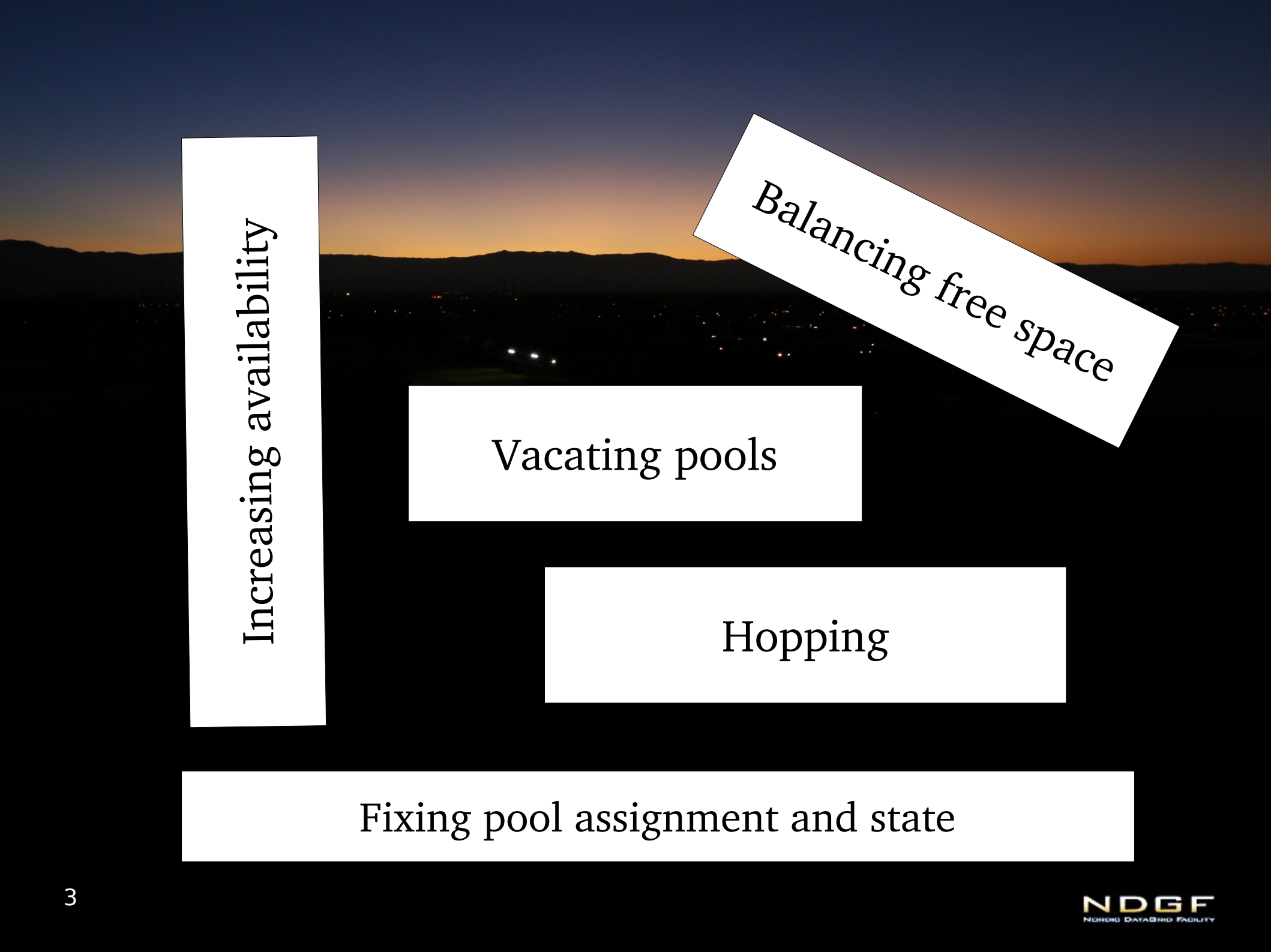
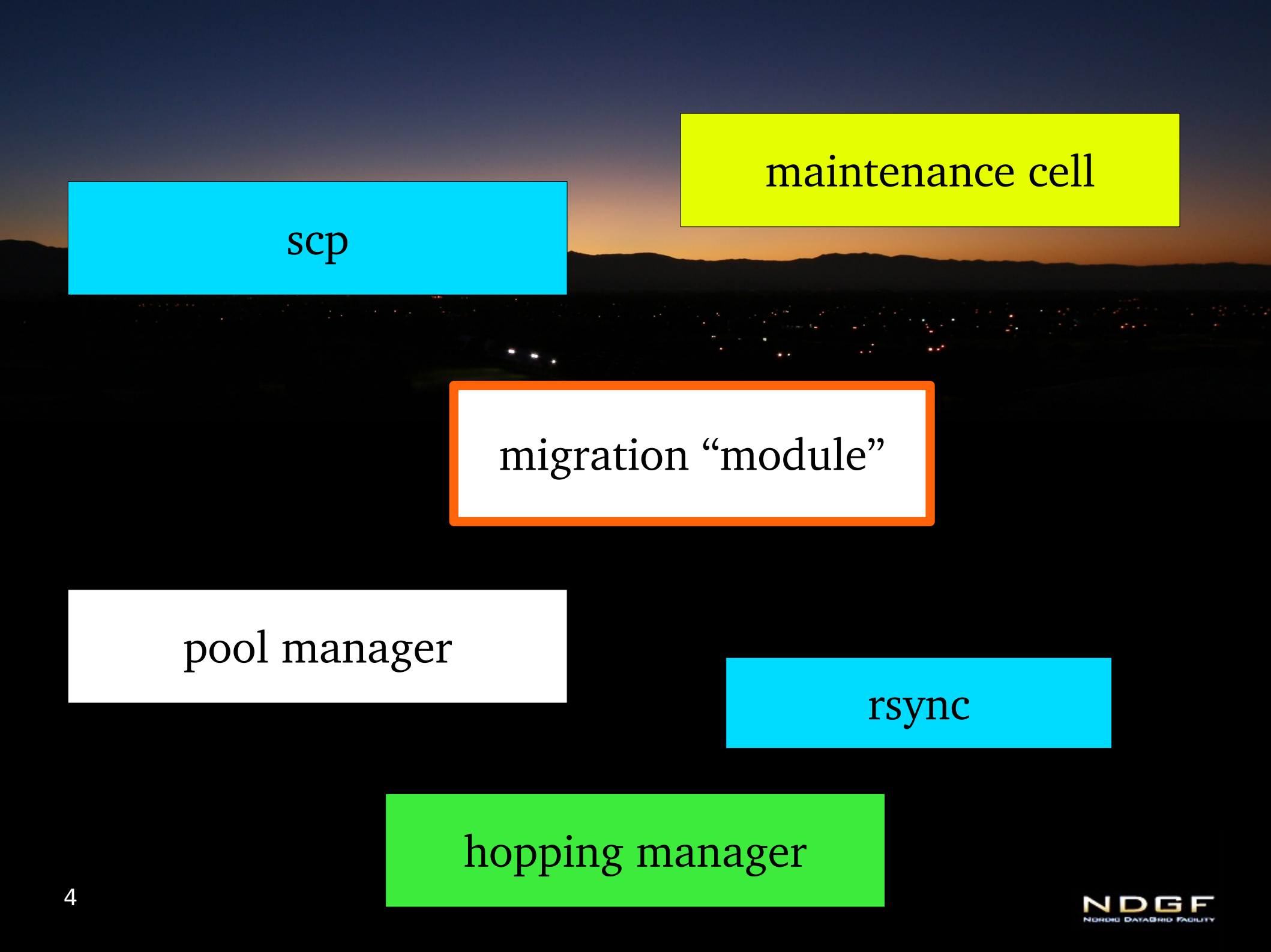
What to remember from this talk

- Migration module is safe
- Migration module can do more than you might know


Increasing availability
Balancing free space
Vacating pools
Hopping
Fixing pool assignment and state

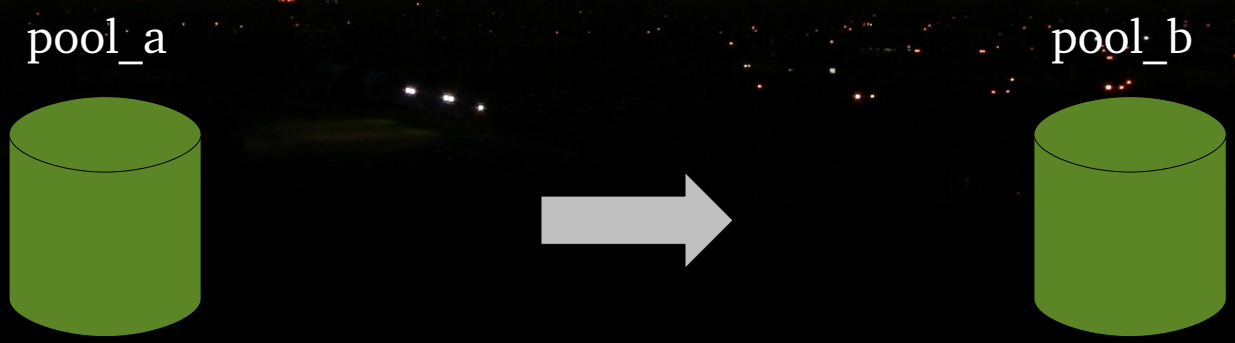

maintenance cell
scp
migration “module”
pool manager
rsync
hopping manager

migration copy pool_b

- Generates exact copy of files on the other pool, including the meta data state.

-accessed=<n>|[<n>]..[<m>]
-al=ONLINE|NEARLINE
-pnfsid=<pnfsid>[,<pnfsid>] …
-rp=CUSTODIAL|REPLICA|OUTPUT

-size=<n>|[<n>]..[<m>]
-state=cached|precious
-sticky[=<owner>[,<owner>…]]
-storage=<class>

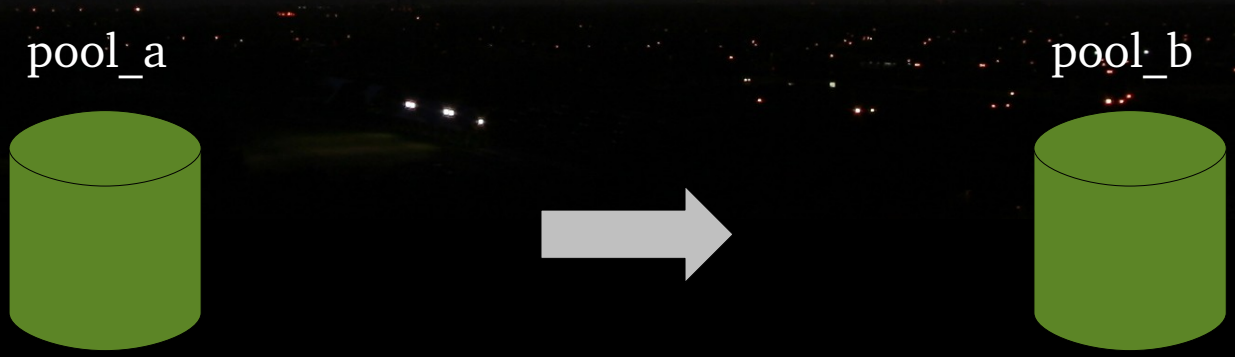


migration copy -state=precious -storage=atlas:default pool_b

- Can filter files on state, sticky flags, storage group, PNFS id, access time, access latency, retention policy, size.

-smode=same|cached|precious|removable|
delete[+<owner>[(<lifetime>)] ...]

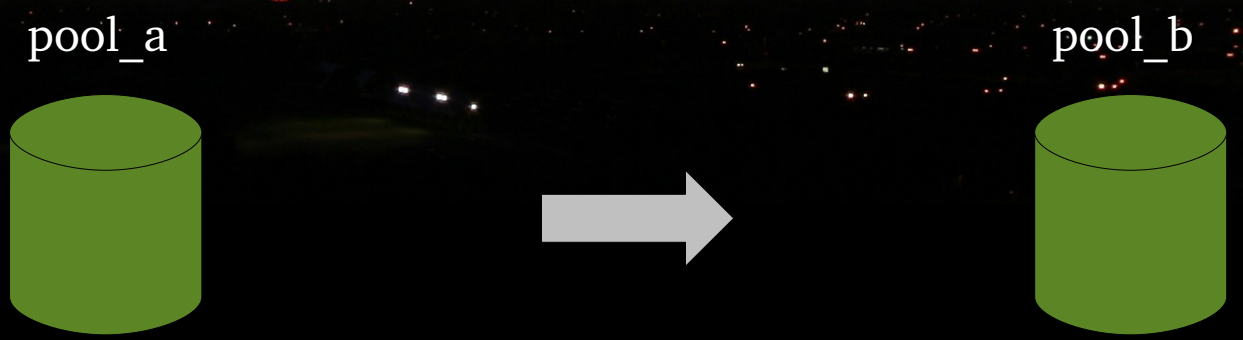
-pins=move|keep



migration copy -smode=removable pool_b

- Can alter the state of the source copy.

-tmode=same|cached|precious[+<owner>[(<lifetime>)] ...]



migration copy -tmode=cached+behrmann(51000) pool_b

- Can set custom state for the target copy.

-target=pool|pgroup|link
-exclude=<pattern>[,<pattern>...]
-include=<pattern>[,<pattern>...]

migration copy -target=pgroup atlas_disk

- Can transfer to pool, set of pools, pool group or link.

-exclude-when=<expr>
-include-when=<expr>


```
migration copy -exclude-when='target.free < 3Ti'
               -target=pgroup atlas_disk
```

- Target set can be dynamic.
  Can refer to size, free, used, removable and total space, cpu cost, space cost, name and both source and target.

-select=proportional|best|random
-refresh=<time>

migration copy -select=random -target=pgroup atlas_disk

- Can balance space usage

-exclude-when=<expr>
-include-when=<expr>
-pause-when=<expr>
-stop-when=<expr>


```
migration copy -target=pgroup atlas_disk
               -exclude-when='target.cpuCost > 0.3'
               -pause-when='source.cpuCost > 0.3 or targets < 4'
```

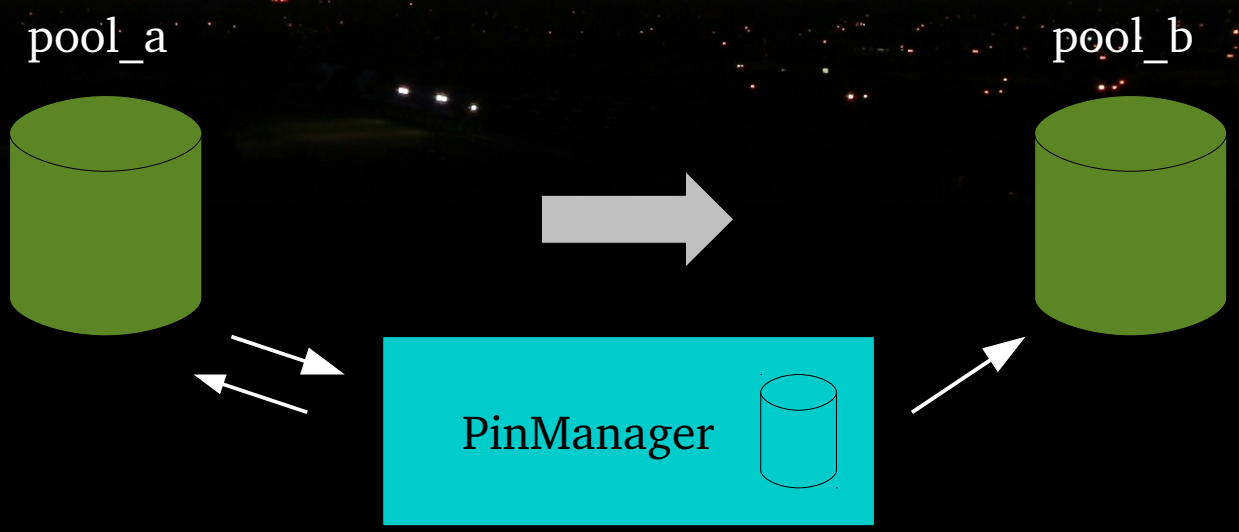
- Adapts to changing conditions.

-stop-when=<expr>
-order=[-]size|[-]lru


```
migration copy -target=pgroup atlas_disk
               -smode=delete
               -stop-when='source.used < 3Ti'
               -order=size
```

- Partial jobs

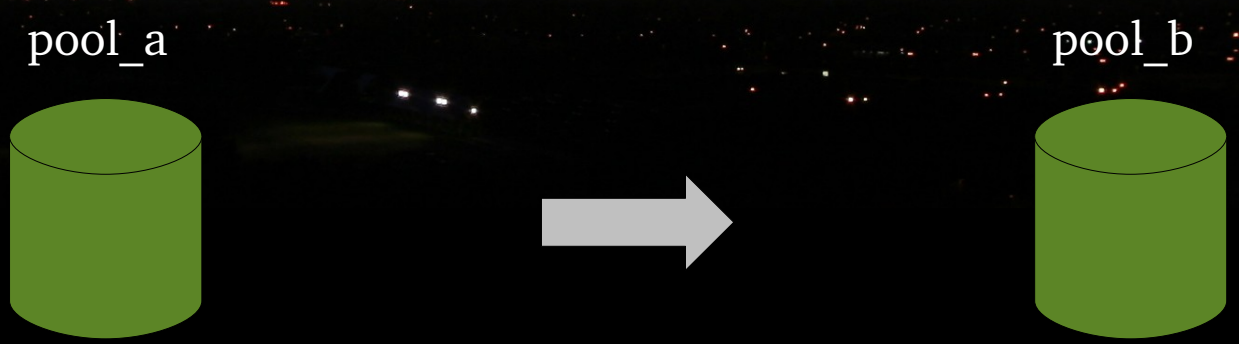
-pins=move|keep



migration copy -pins=move pool_b

- Integrates with pin manager to move pins.
  Never deletes a pin.
  Never deletes a file if there is still a pin on it.

csm set policy [-onwrite=on|off] [-ontransfer=on|off]



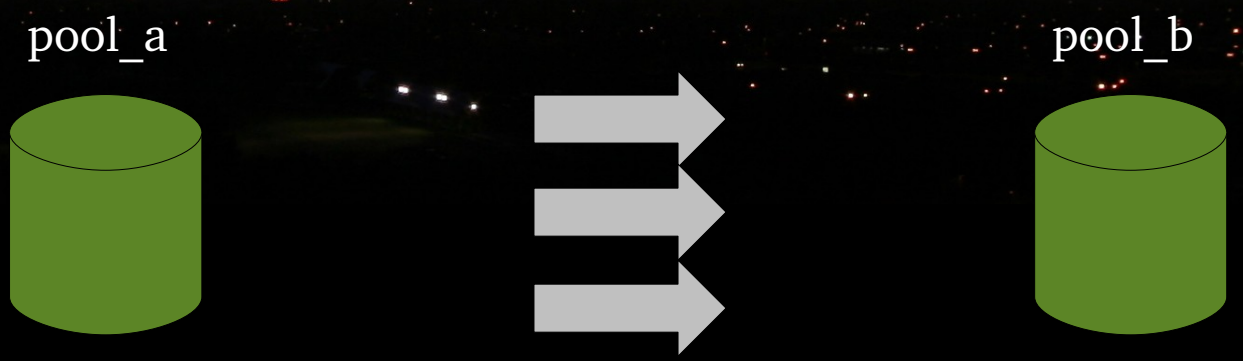
migration copy pool_b

csm set policy -onwrite=off
-ontransfer=on

- Subject to checksum policy on target pool

-concurrency=<concurrency>

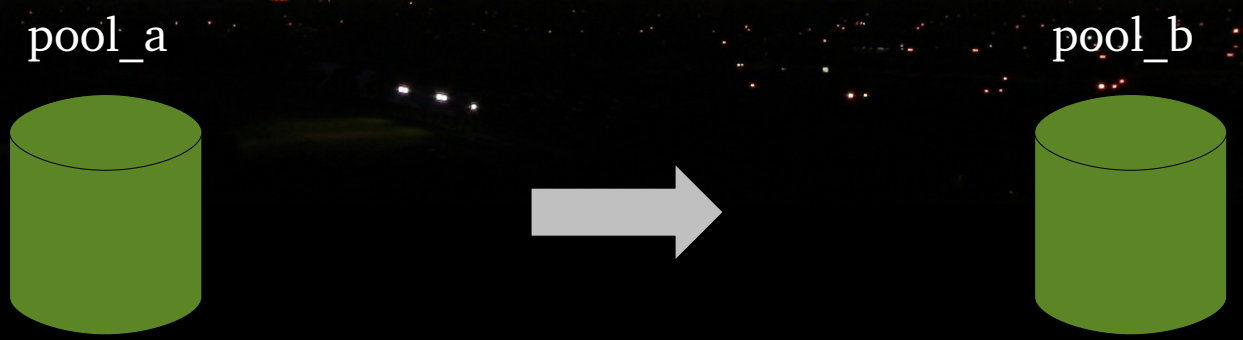
migration concurrency <job> <n>



migration copy -concurrency=3 pool_b

- Multiple concurrent transfers.

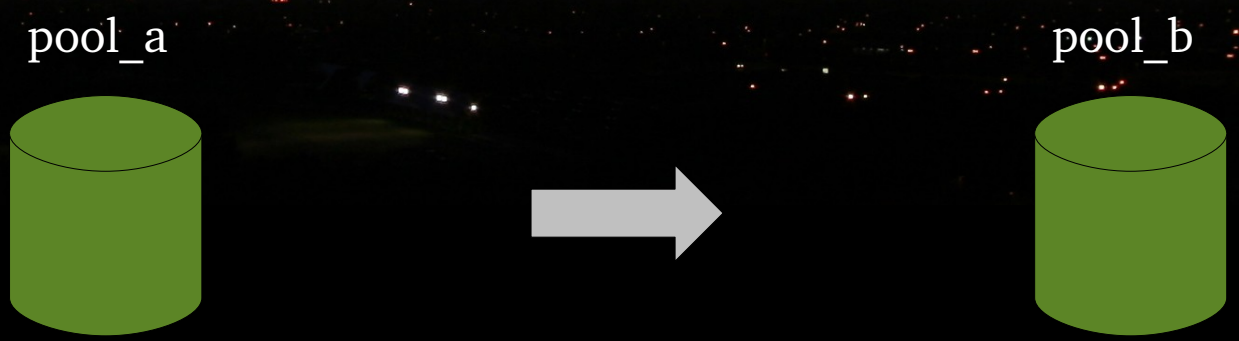
migration move ….
migration cache ….



migration move pool_b = migration -smode=delete -tmode=same -pins=move -verify pool_b

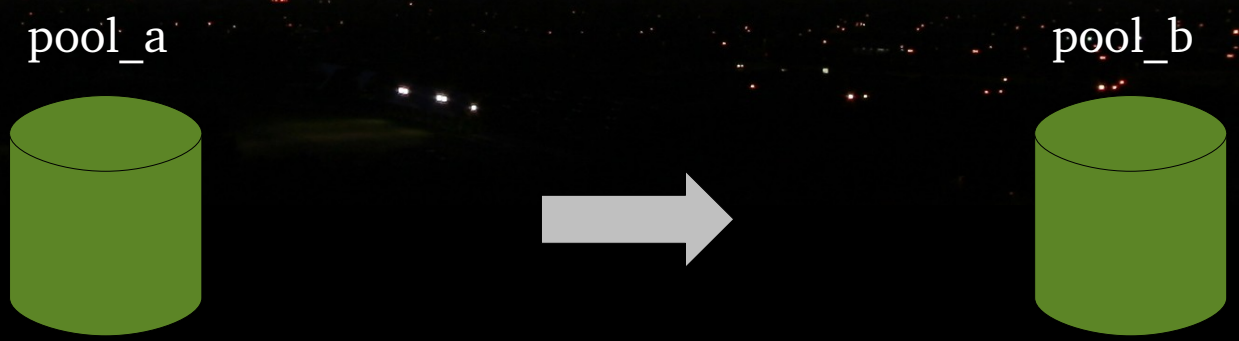
- Convenient short hands for moving and caching files.
  Only difference is the default values.

-permanent



migration move -al=CUSTODIAL -rp=CUSTODIAL
-permanent pool_b

- Permanent jobs never terminate.
  Are saved to setup file with save command.



migration copy pool_b → pp get file <pnfsid> pool_a

- Actual transfer is a regular pool to pool transfer.

migration copy -target=pgroup atlas_disk
migration copy -target=pgroup atlas_disk

=

migration copy -target=pgroup atlas_disk

- Operation is **idempotent**!
  Can be rerun or restarted without changing the result.

Source	Existing target	New target
Cached	Cached	Cached
Cached	Precious	Precious
Precious	Cached	Precious
Precious	Precious	Precious
sticky(n)	sticky(m)	sticky(max(n,m))

-verify

migration copy -smode=delete -verify -target=pgroup atlas_disk

- Can optionally verify the checksum of an existing target.
  On by default when using the move command.

-eager

migration copy -eager -target=pgroup atlas_disk

- Usually waits for offline pools to become online, but an eager job creates new copies instead.

rebalance pgroup [-metric=relative|sc] [-refresh=<period>] <pgroup>
rebalance cancel pgroup <pgroup>

rebalance pgroup -metric=relative atlas_disk

EXPERIMENTAL

- PoolManager can submit migration jobs to all pools in a pool group to balance space usage.

What to remember from this talk

- Migration module is safe

  Uses the existing pool to pool transfer code.
  Extensive checksum validation.

  Idempotent operations allow jobs to fail, to be cancelled and to be restarted without ill effects.

  Never touches pins – relies on PinManager to move pins.

  Safe doesn't mean you can't shoot yourself in the foot!

What to remember from this talk

- Migration module can do more than you might know

  Copying, moving, caching
  Filtering
  State manipulation
  Permanent jobs
  Advanced and dynamic selection criteria
  Adapts to load conditions
  Doesn't have to transfer everything