



Neues von RMS


RMS New Features: Ein Review

2G03

Thilo Lauer
thilo.lauer@hp.com

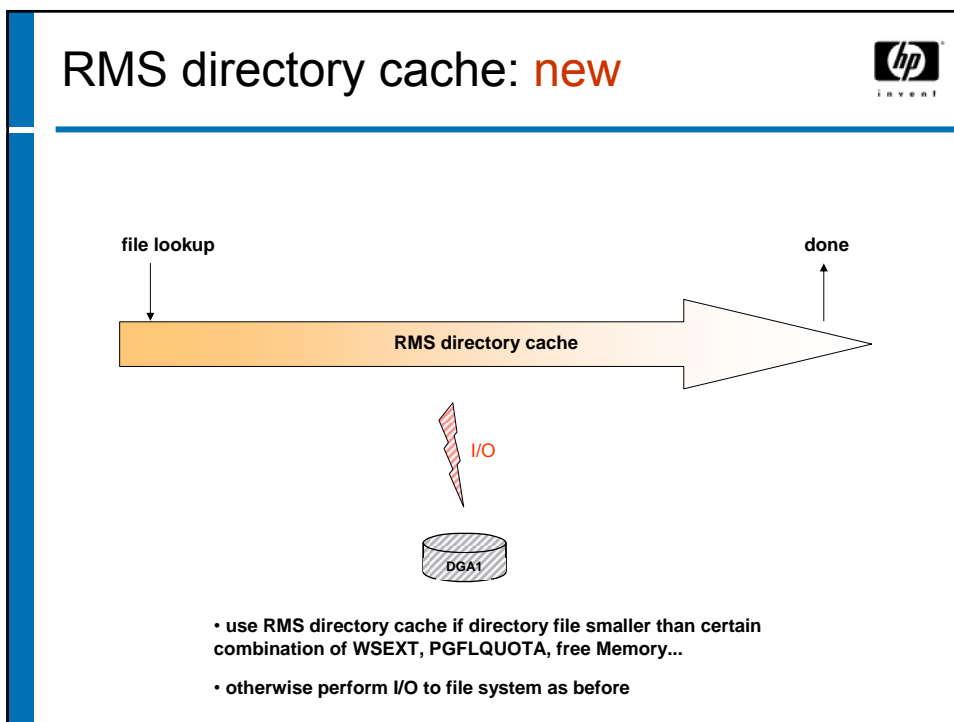
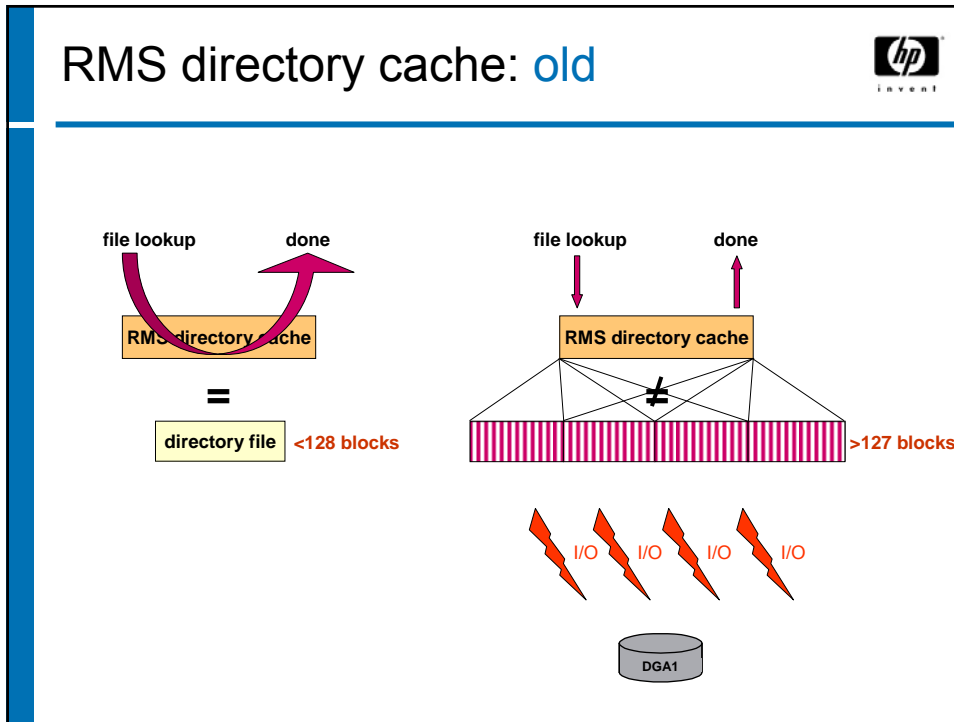


© 2005 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice



Areas of concern

- RMS directory cache V7.2
- global buffer lookup algorithm V7.2
- access to global buffer cache V7.2
- global buffer bucket locking V7.2-1H1
- no-locked read access V7.2-1H1
- locking fairness issues V7.2-1H1
- global buffer size/number limits V8.3 ~~???~~
 - new global buffer management options

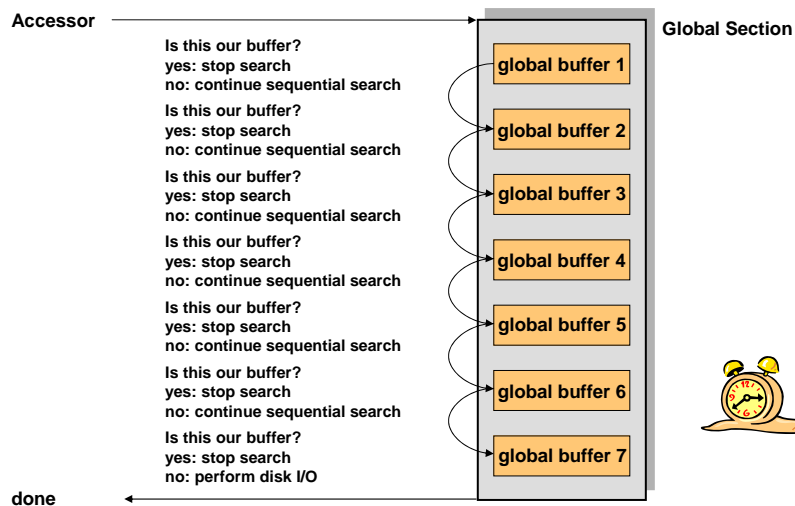


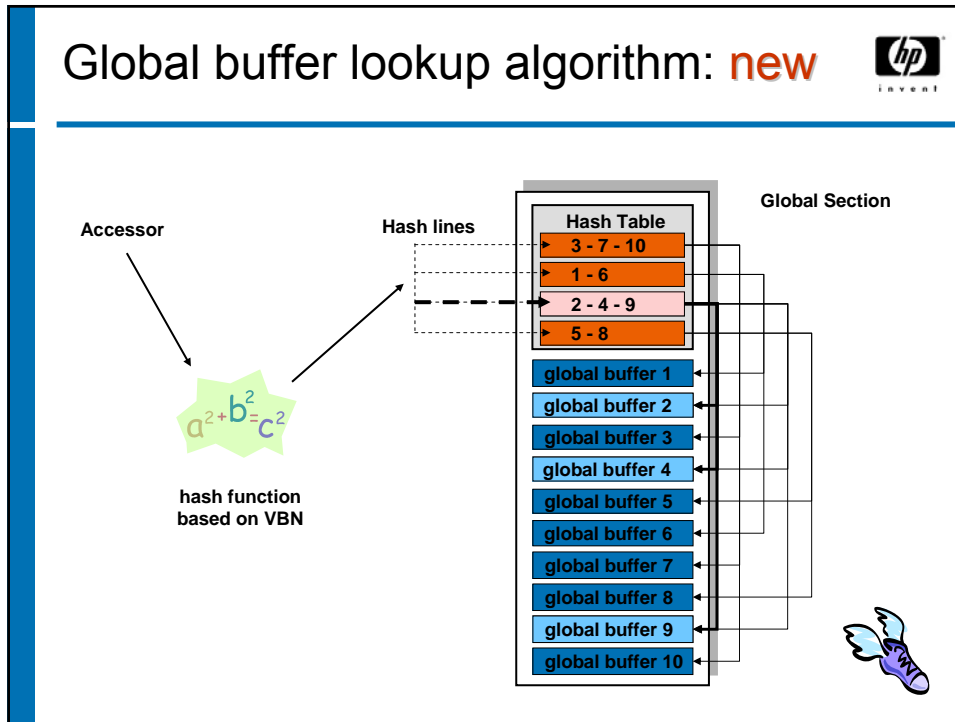
RMS directory cache: effects



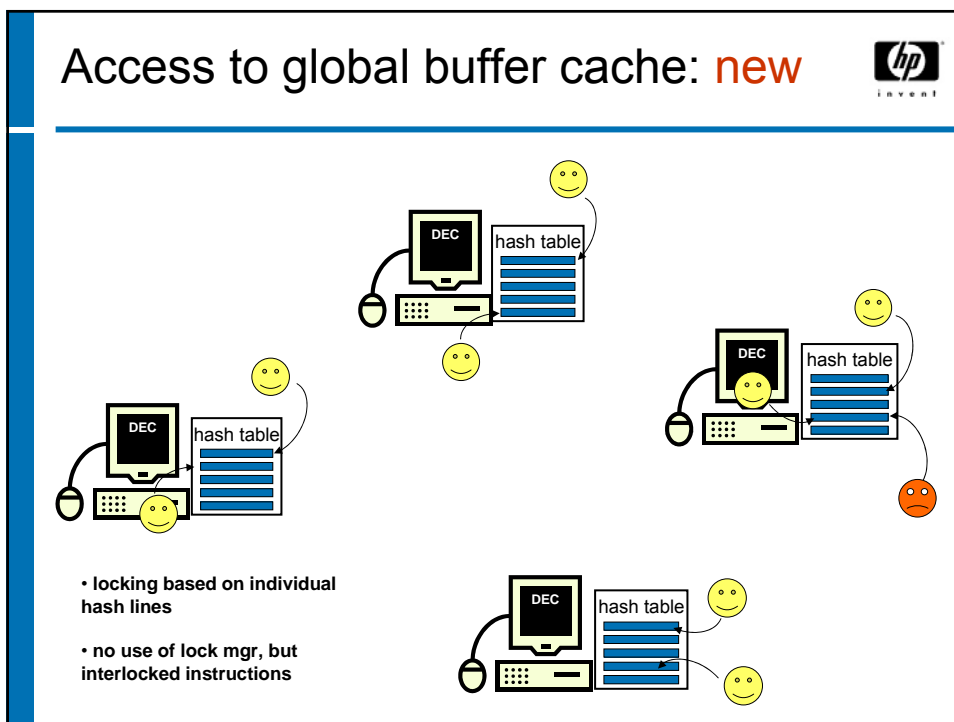
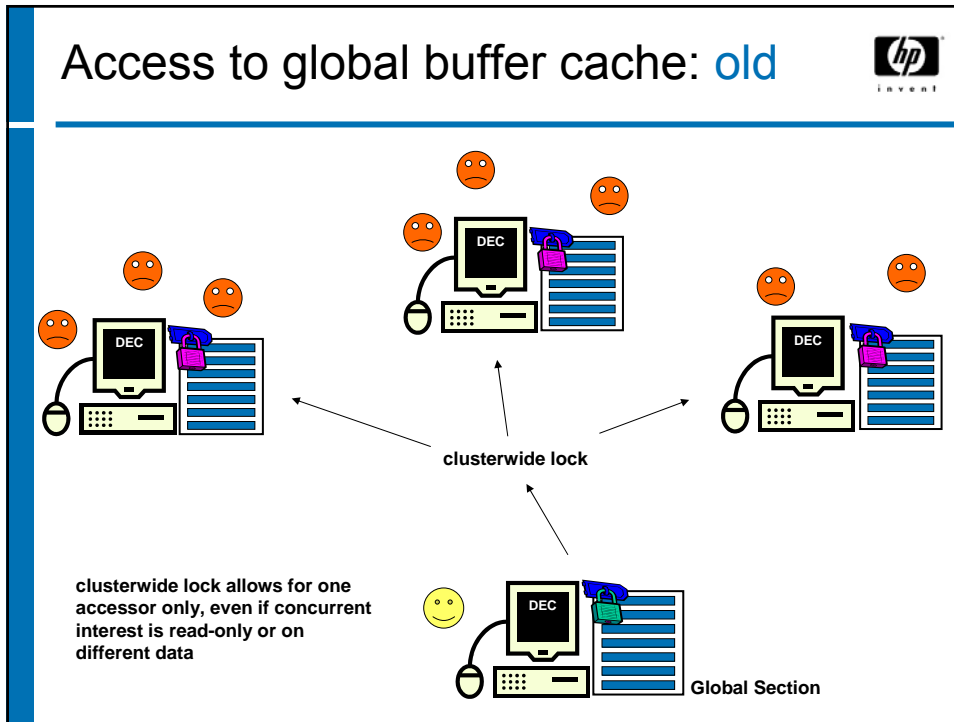
- no File System Operations involved
- fewer or no I/Os during wildcard directory lookup
- less CPU overhead

Global buffer lookup algorithm: old





- ## Global buffer lookup algorithm: **effects**
- GB lookup scales better:
 - old: $N/2$
 - new: between $N/(\text{bucket size})$ and $N/(2 * \text{bucket size})$
 - provides for much larger GB settings
 - no change in absolute limits
 - but higher practical values are now reasonable
 - small increase in size of global section



Access to global buffer cache. effects

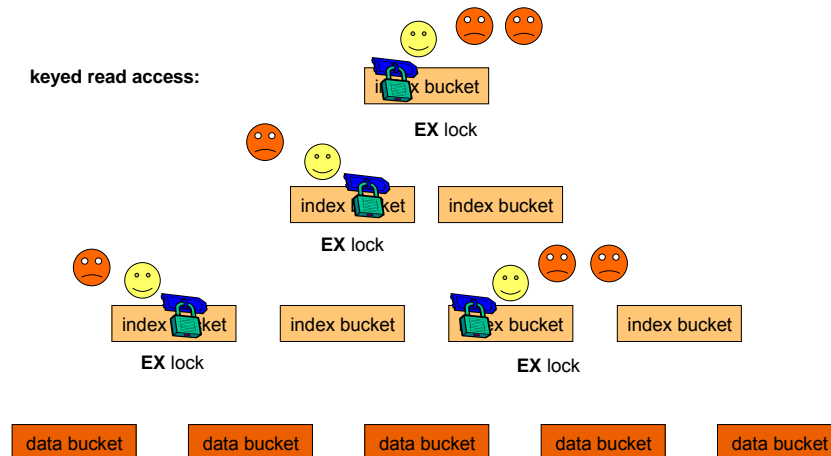


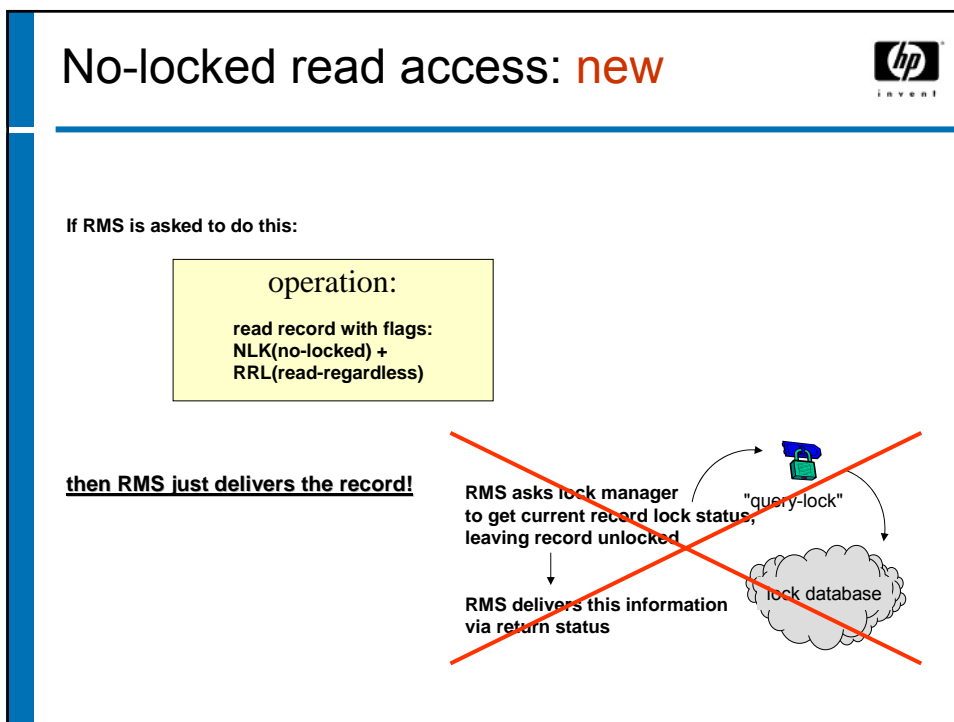
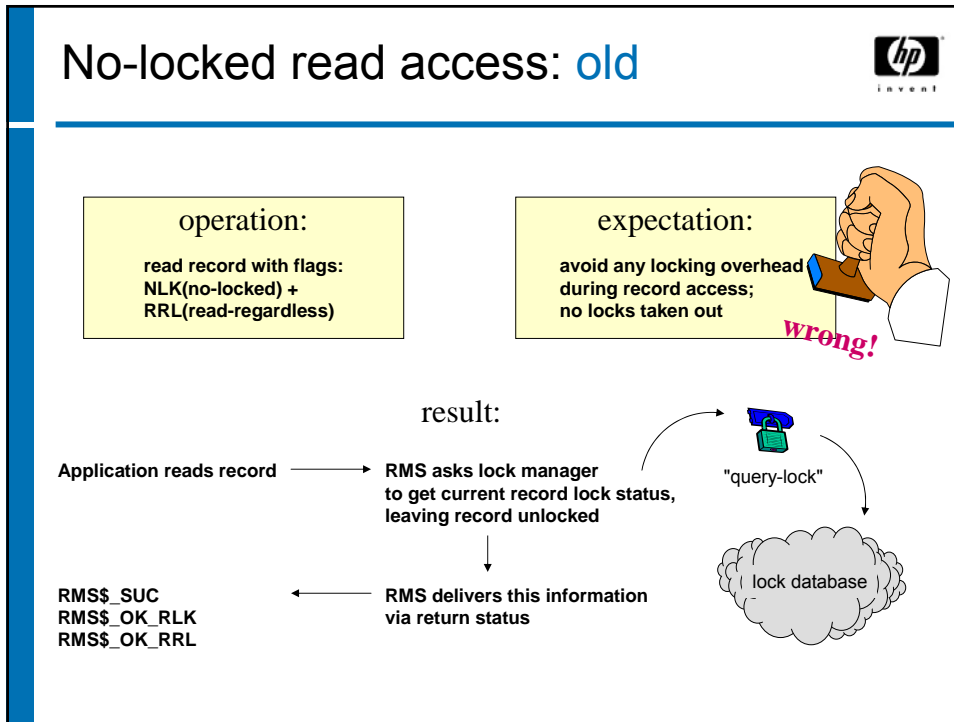
- higher locking granularity
- concurrent access on global section
- higher overall throughput
- no clusterwide locking traffic
- less CPU load

Global buffer bucket locking: old



keyed read access:



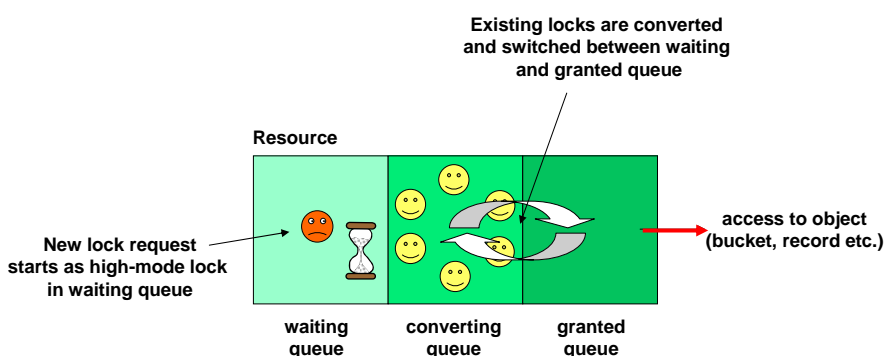


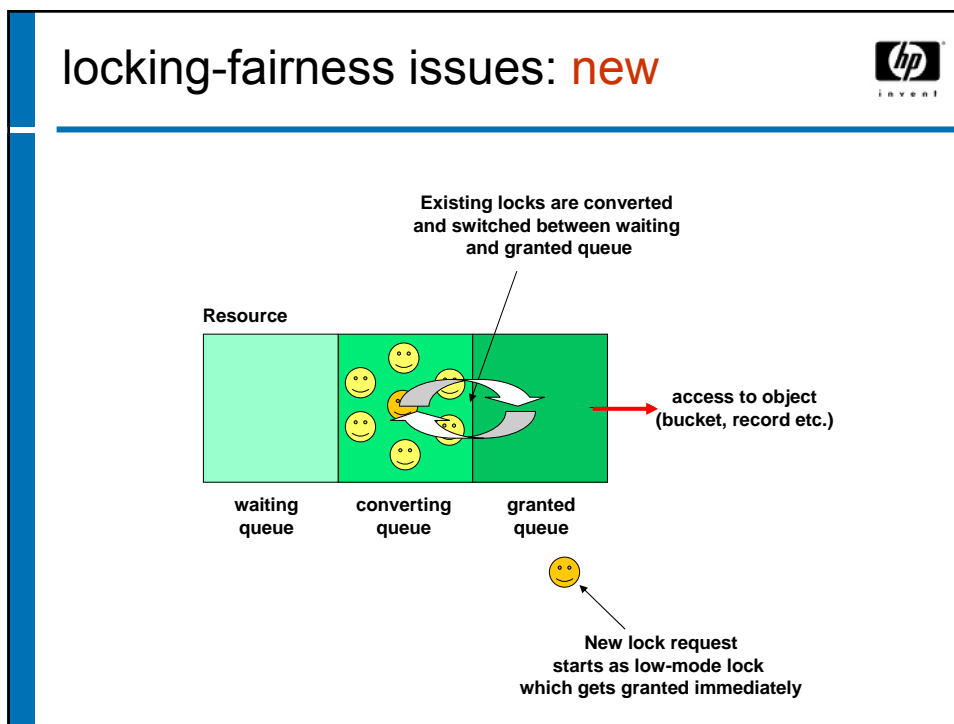
No-locked read access: effects



- true no-locked read access
- no locking overhead at all:
 - read of N records needs 0 instead of N locks
- Some customer applications run 30% faster!
- **Beware:** Application doesn't get any information on locking state of record. Check your code!
- Feature is accessible via \$ SET RMS. No code change necessary.
- Also I/O-based control via flag (NQL in FAB or RAB)

locking-fairness issues: old





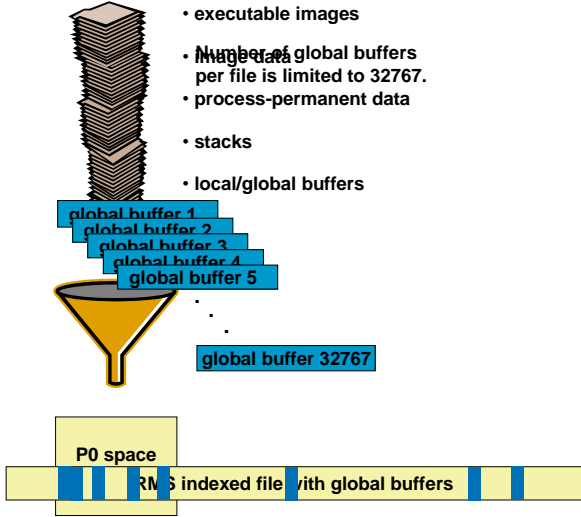
- ## locking-fairness issues: **effects**
- previous implementation had potential for locking out new accessors for extended periods of time
 - new impl. provides more fairness between accessors
 - all accessors are treated equally, 'democratic' lock usage
 - better average response time

global buffer size/number limits: old

pre-V8.3

Global buffers reside in P0 space, which is limited and shared with others

- executable images
- ~~Number of~~ global buffers per file is limited to 32767.
- process-permanent data
- stacks
- local/global buffers



P0 space

Indexed file with global buffers

global buffer size/number limits: new

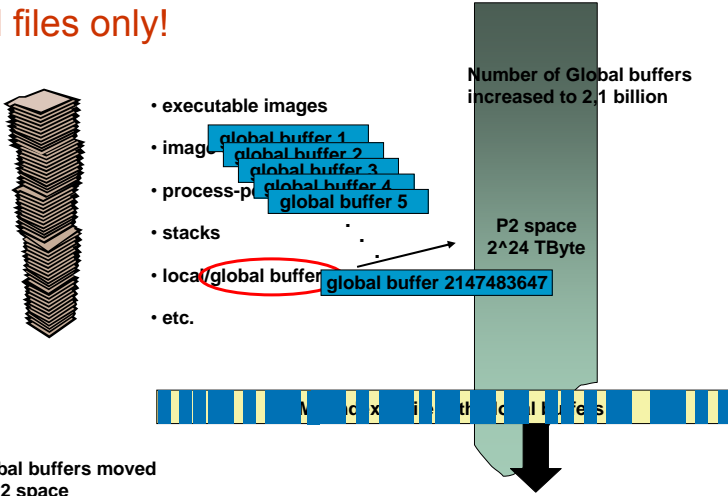
V8.3 (V8.2 on special request)

Indexed files only!

- executable images
- image global buffer 1
- global buffer 2
- process-p global buffer 3
- global buffer 4
- global buffer 5
- stacks
- local/global buffer global buffer 2147483647
- etc.

Number of Global buffers increased to 2.1 billion

P2 space
2²⁴ TByte



Indexed files

Global buffers moved to P2 space

New global buffer mgmt. options...



Valid for all file organizations:

- old „Count“ option
 - `$SET FILE/GLOBAL_BUFFERS=1000`
 - `$SET FILE/GLOBAL=COUNT=1000000`
- new „Percent“ option
 - `$SET FILE/GLOBAL=PERCENT=35`
calculate buffers during first open of a file on given node
- new „Default“ option
 - `$SET FILE/GLOBAL=DEFAULT`
cache whole file if used blocks < GB_CACHEALLMAX
cache GB_DEFPERCENT% if used blocks > GB_CACHEALLMAX

...New global buffer mgmt. options...

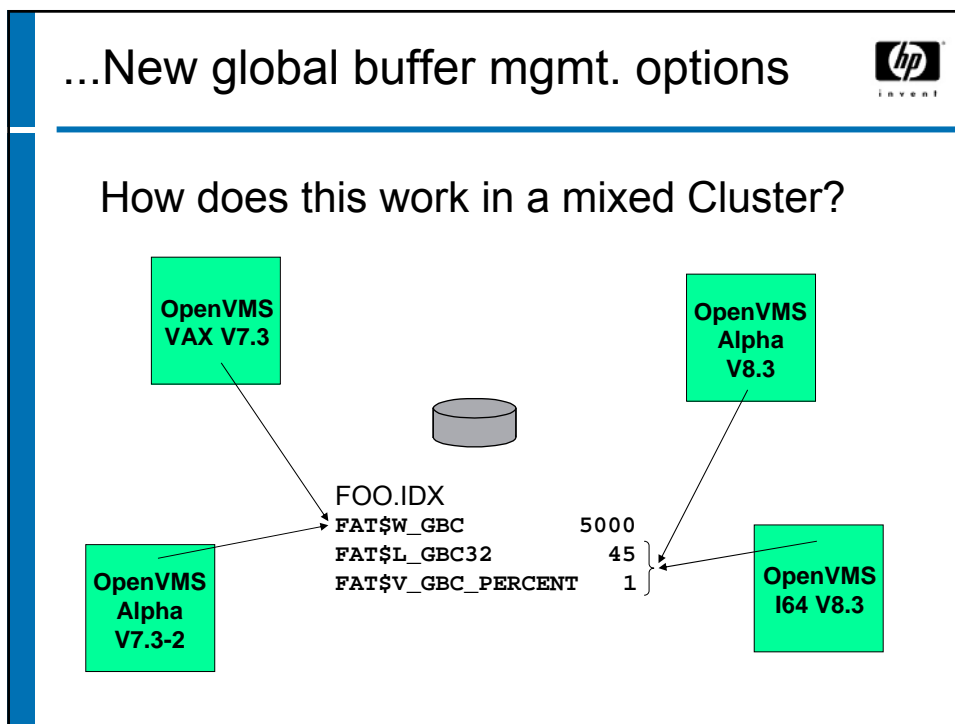


DIRECTORY/FULL:

- Traditional setting
`"Global buffer count: 5000"`
- Use of new options
`"Global buffer count pre-V8.3: 5000
Global buffer count V8.3+ : 45 percent"`

ANALYZE/RMS:

```
"Global Buffer Count pre-V8.3: 5000  
Global Buffer Count V8.3+ : 45  
Global Buffer Flags V8.3+ : percent"
```



Questions...

thilo.lauer@hp.com
thilo.lauer@compaq.com
thilo.lauer@digital.com