Hugetlbfs Still Alive and Kicking

Mike Kravetz, Oracle Corporation

The opinions expressed in this presentation are the presenter's own, and do not represent the views of Oracle or anyone else.

THE LINUX FOUNDATION

Why Huge Pages?

- Virtual memory == address translations
- TLB cache for quick lookups
- One page per TLB entry
- Bigger pages can mean fewer TLB misses

Huge Pages in Linux

- hugetlbfs
  - The oldest method
  - System config and application changes
  - Large databases early (and current) users
- Transparent Huge Pages (THP)
  - As the name says ‘transparent’
  - Used without sys config or app changes

hugetlbfs history

- 2.5.36 kernel, September 2002
  - Followed the everything is a file model
  - Explicitly mount filesystem
    - open and mmap files
- shmget(SHM_HUGETLB)
- mmap(MAP_HUGETLB)

Reservations

- page faults allocate hugetlbfs pages
- ENOSPC / ENOMEM == SIGBUS
- shmget() and mmap() reserve pages
  - ENOMEM on error, faults guaranteed

```
# grep HugePages /proc/meminfo

HugePages_Total:    1024
HugePages_Free:     1024
HugePages_Rsvd:      512
HugePages_Surp:        0
```

Surplus Pages and Overcommit

- huge pages commonly preallocated
  - kernel command line
  - System init scripts
- nr_overcommit_hugepages
  - dynamically allocated
    - mmap() or page fault time
    - no guarantees
  - surplus_hugepages

Multiple Huge Page Sizes

- x86: 2M and 1G
- Powerpc: 512K, 1M, 2M, 8M, 16M, 1G, 16G
- Default huge page size
  - Shown in /proc/meminfo
  - Set on kernel command line
- /sys/kernel/mm/hugepages (system supp)
- shmget() and mmap() take size

libhugetlbfs

- An ‘easy’ way to use hugetlbfs
- library controlled by environment variables
- malloc, System V shared memory
  - LD_PRELOAD and no application mods
- Program, text, data and bss
  - Relink program for optimal benefit
  - Segments ‘copied’ at program startup

Shared Page Tables

- Yes, PMDs can be shared: **IF**
  - You are using huge pages on x86 or arm64
  - Of sufficient size and alignment: PUD (1G)
- Example: 1TB shared mapping, 10,000 mappings
  - 4KB (PMD) per 1G
  - 1TB = 1024G * 9,999 shared mappings
  - Up to 39GB savings

Mount reservations

- Added in v4.1
- huge pages reside in global pools
- Can be taken by any user/program (WSP)
- filesystem mounts can reserve pages
  - min_size=<value> mount option

fallocate support

- Added in v4.3
- preallocating pages is not exciting
- deallocating (hole punch) is interesting
  - Can ‘release’ pages in hugetlbfs files
  - madvise(MADV_REMOVE) if no fd

userfaultfd support

- Added in v4.11
- Catch access to fallocate holes (orig UC)
- QEMU post copy live migration (new UC)
  - Now works with huge pages
  - Performance boost for 2M pages
  - GB pages too much network latency

Questions?

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

NORTH AMERICA

THE LINUX FOUNDATION