Linaro
connect
Budapest 2017

Implementing contiguous page hint for anonymous pages in user space

Bamvor Jian Zhang

Agenda

- Page size and performance
- Contiguous page hint
- Current usage
- Our Idea

BIO

- Kernel developer in kwg
- Focus on ILP32 in recent two years
- Work cont page hint recently
- Presentation:
  - 2014 Opensuse Asia Summit: openSUSE on ARM
  - 2016 Linuxcon Europe: An efficient unit test and fuzz tools for kernel/libc porting
  - 2016 Linaro Connect Las Vegas: LAS16-TR07: Working upstream [Mandarin]

The bottleneck of memory

- Fragmentation
- Latency
- High performance memory usage

Increasing the page size?

- 64k base pages is probably not a good idea
  - One order of magnitude higher memory use with 64k pages.
  - I/O amplification
- Page size performance measurements
  - There is no overall improvement for filesystem.
- Specint, Why?
  - Care about system benchmark other than micro benchmark
  - not overly affected by wasted memory or I/O performance
  - sensitive to TLB misses
- Specint, result
  - There is no overall improvement when we change the page size from 4k to 64k
  - Some of test cases downgrade: hmmer, xalancbmk.

Compare the performance between 4k and 64k

	4k without THP	4k with THP	64k with THP disable	64k with THP enable
400.perlbench	100%	101.59%	102.38%	102.38%
401.bzip2	100%	100.53%	102.88%	103.21%
403.gcc	100%	101.58%	103.16%	103.29%
429.mcf	100%	119.65%	117.26%	118.33%
445.gobmk	100%	100.88%	101.77%	101.77%
456.hmmer	100%	100.00%	60.39%	59.67%

Linaro
connect
Budapest 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Compare the performance between 4k and 64k

	4k without THP	4k with THP	64k without THP	64k with THP
458.sjeng	100%	102.88%	103.85%	101.92%
462.libquantum	100%	105.88%	109.80%	114.38%
471.omnetpp	100%	112.54%	113.04%	112.04%
473.astar	100%	108.59%	110.59%	109.76%
483.xalancbmk	100%	108.11%	105.41%	106.31%

Contiguous page hint

- Support armv7-a and armv8-a.
- Place hint in page table if contiguous pages
- Could save TLB entries (could, not must) and decrease the tlb miss accordingly

Contiguous page hint: configuration

Page size	level	Number of continuous entries	size
4k	pmd	16	32M
4k	pte	16	64K
16k	pmd	32	1G
16k	pte	128	2M
64k	pmd	32	16G
64k	pte	32	2M

Current usage

- Kernel mem
  - emulate 2M hugetlb in 64k page
- Filesytem
  - bb9f96b
- virtualization
  - Place cont page hint for xen hypervisor

Some thoughts for user space

- Use hugetlb directly?
- Maintain the 16page all the time?
- Lazy page hint set and split when needed?

The relationship between performance and tlb miss

	performance	Dtlb load miss
462.libquantum	103.92%	57.81%
473.astar	102%	66.2%

Some thoughts for user space

- Use hugetlb directly?
- Maintain the 16page all the time?
- Lazy page hint set and split when needed?

Some thoughts for user space

- Use hugetlb directly?
- Maintain the 16page all the time?
- Lazy page hint set and split when needed?

Our idea

- Allocate the continuous 64k pages in the first time of fault
    - It is after the THP and hugetlb handle.
- Set all the pte and cont page hint in the second fault of same region
- When next fault happens in another region, free all reserved pages
- Split the 64k page when necessary

Reference

- https://www.usenix.org/system/files/conference/osdi16/osdi16-kwon.pdf

Linaro
connect
Budapest 2017

Thank You

#BUD17
bamvor.zhangjian@linaro.org/bamv2005@gmail.com
For further information: www.linaro.org
BUD17 keynotes and videos on: connect.linaro.org