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

<table>
<thead>
<tr>
<th>Test</th>
<th>4k without THP</th>
<th>4k with THP</th>
<th>64k with THP disable</th>
<th>64k with THP enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>400.perlbench</td>
<td>100%</td>
<td>101.59%</td>
<td>102.38%</td>
<td>102.38%</td>
</tr>
<tr>
<td>401.bzip2</td>
<td>100%</td>
<td>100.53%</td>
<td>102.88%</td>
<td>103.21%</td>
</tr>
<tr>
<td>403.gcc</td>
<td>100%</td>
<td>101.58%</td>
<td>103.16%</td>
<td>103.29%</td>
</tr>
<tr>
<td>429.mcf</td>
<td>100%</td>
<td>119.65%</td>
<td>117.26%</td>
<td>118.33%</td>
</tr>
<tr>
<td>445.gobmk</td>
<td>100%</td>
<td>100.88%</td>
<td>101.77%</td>
<td>101.77%</td>
</tr>
<tr>
<td>456.hmmer</td>
<td>100%</td>
<td>100.00%</td>
<td>60.39%</td>
<td>59.67%</td>
</tr>
</tbody>
</table>
## Compare the performance between 4k and 64k

<table>
<thead>
<tr>
<th>Test Case</th>
<th>4k without THP</th>
<th>4k with THP</th>
<th>64k without THP</th>
<th>64k with THP</th>
</tr>
</thead>
<tbody>
<tr>
<td>458.sjeng</td>
<td>100%</td>
<td>102.88%</td>
<td>103.85%</td>
<td>101.92%</td>
</tr>
<tr>
<td>462.libquantum</td>
<td>100%</td>
<td>105.88%</td>
<td>109.80%</td>
<td>114.38%</td>
</tr>
<tr>
<td>471.omnetpp</td>
<td>100%</td>
<td>112.54%</td>
<td>113.04%</td>
<td>112.04%</td>
</tr>
<tr>
<td>473.astar</td>
<td>100%</td>
<td>108.59%</td>
<td>110.59%</td>
<td>109.76%</td>
</tr>
<tr>
<td>483.xalancbmk</td>
<td>100%</td>
<td>108.11%</td>
<td>105.41%</td>
<td>106.31%</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Page size</th>
<th>level</th>
<th>Number of continuous entries</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4k</td>
<td>pmd</td>
<td>16</td>
<td>32M</td>
</tr>
<tr>
<td>4k</td>
<td>pte</td>
<td>16</td>
<td>64K</td>
</tr>
<tr>
<td>16k</td>
<td>pmd</td>
<td>32</td>
<td>1G</td>
</tr>
<tr>
<td>16k</td>
<td>pte</td>
<td>128</td>
<td>2M</td>
</tr>
<tr>
<td>64k</td>
<td>pmd</td>
<td>32</td>
<td>16G</td>
</tr>
<tr>
<td>64k</td>
<td>pte</td>
<td>32</td>
<td>2M</td>
</tr>
</tbody>
</table>
Current usage

- **Kernel mem**
  - emulate 2M hugetlb in 64k page

- **Filesysytem**
  - 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

<table>
<thead>
<tr>
<th></th>
<th>performance</th>
<th>Dtlb load miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>462.libquantum</td>
<td>103.92%</td>
<td>57.81%</td>
</tr>
<tr>
<td>473.astar</td>
<td>102%</td>
<td>66.2%</td>
</tr>
</tbody>
</table>
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 16 page 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

Thank You

#BUD17

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