Binder Driver

Contention in buffer allocations

Carlos Llamas | cmllamas@google.com
LPC Richmond, Virginia | 2023
Problem
Binder model

- Service
  - mmap()
  - Thread Pool
    - Thread 1
    - Thread 2
    - Thread 3
    - Thread 4
    - Thread Max

- Clients
  - Client 1
  - Client 2
  - Client 3
  - Client n
  - ...

- Proprietary + Confidential
Problem - Janks due to contention

- Requests are serialized via binder alloc->mutex
- Reports of major source of contention
- Low-priority task sleeps with alloc->mutex
- mmap lock was nested under alloc->mutex
- Memory pressure increase contention
Solution
Solution - Mitigate the contention

- Extract operations that `might_sleep()` from `alloc->mutex`
- **Split** the new buffer setup from the page insertion routine
- No longer have `mmap_lock` **nested** under `alloc->mutex`
- Convert the `alloc->mutex` into a **spinlock**
- **v1** submitted upstream [here](#)
Results

- From a stress test and measured **wall-time**

<table>
<thead>
<tr>
<th>sched</th>
<th>prio</th>
<th>average</th>
<th>max</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>fifo</td>
<td>99</td>
<td>0.350ms</td>
<td>248.730ms</td>
<td>0.020ms</td>
</tr>
<tr>
<td>fifo</td>
<td>01</td>
<td>0.357ms</td>
<td>248.817ms</td>
<td>0.024ms</td>
</tr>
<tr>
<td>other</td>
<td>-20</td>
<td>0.399ms</td>
<td>249.906ms</td>
<td>0.020ms</td>
</tr>
<tr>
<td>other</td>
<td>19</td>
<td>0.477ms</td>
<td>297.756ms</td>
<td>0.022ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sched</th>
<th>prio</th>
<th>average</th>
<th>max</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>fifo</td>
<td>99</td>
<td>0.135ms</td>
<td>1.197ms</td>
<td>0.022ms</td>
</tr>
<tr>
<td>fifo</td>
<td>01</td>
<td>0.136ms</td>
<td>5.232ms</td>
<td>0.018ms</td>
</tr>
<tr>
<td>other</td>
<td>-20</td>
<td>0.180ms</td>
<td>7.403ms</td>
<td>0.019ms</td>
</tr>
<tr>
<td>other</td>
<td>19</td>
<td>0.241ms</td>
<td>58.094ms</td>
<td>0.018ms</td>
</tr>
</tbody>
</table>

200x faster
Q&A
Rust binder rewrite

- Authored by Wedson Almeida Filho [WIP]
- Alice Ryhl submitted an RFC to lkml [here]
- Reduces complexity and memory bugs
- Currently testing the experiment
- Goal is feature and performance parity
Thank You