BRAVO – Biased Locking for Reader-Writer Locks

Dave Dice  Alex Kogan
Oracle Labs
Reader-Writer Locks

- Allow shared access for read-only use of a resource
- Ubiquitous in modern systems
Reader-Writer Locks

• Allow shared access for read-only use of a resource
• Ubiquitous in modern systems

• Have to keep track of the presence of active readers
The “shared counter” approach
The “shared counter” approach
The “shared counter” approach
The “shared counter” approach
The “shared counter” approach
The “distributed” approach
The “distributed” approach
The Scalable “Reader Indicator” Dilemma

Compact  Scalable
The BRAVO approach

Compact & Scalable
The BRAVO approach
The BRAVO approach

0 → 0

RBias
Inhibit Until
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach

now + t * 10 = t
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
The BRAVO approach
Evaluation

• Easy to integrate with existing locks

• Compact

• Accelerates reads

• Handles writes gracefully
Evaluation: Easy to integrate

- Brandenburg-Anderson (BA) reader-writer lock
- POSIX Pthread reader-writer lock
- Linux kernel rwsem
Evaluation: Compact

<table>
<thead>
<tr>
<th>Locks</th>
<th>Memory footprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA</td>
<td>40</td>
</tr>
<tr>
<td>BA + BRAVO</td>
<td>40 + 12 + 32KB (for a table)</td>
</tr>
<tr>
<td>Per-CPU</td>
<td>9216 (on a system with 72 CPUs)</td>
</tr>
<tr>
<td>Cohort-RW</td>
<td>896 (dual-socket)</td>
</tr>
</tbody>
</table>

Intel Xeon E5-2699 v3 CPU
2 sockets
72 logical CPUs in total
Evaluation: Accelerates reads
RWBench with 1 out of every 10000 are writes
Evaluation: Handles writes gracefully
RWBench with 9 out of every 10 are writes

![Diagram showing aggregate throughput rate (ops/msec) vs threads]

- Cohort-RW
- Per-CPU
- BA
- BRAVO-BA
- pthread
- BRAVO-pthread
Linux Kernel rwsem

• Counter + waiting queue protected by a spin lock

• Reader atomically increments the counter and checks its value
Linux Kernel rwsem

• Counter + waiting queue protected by a spin lock

• Reader atomically increments the counter and checks its value

• Synchronization bottleneck in the kernel (mmap_sem)
Linux Kernel rwsem

• Counter + waiting queue protected by a spin lock

• Reader atomically increments the counter and checks its value

• Synchronization bottleneck in the kernel (mmap_sem)

• Stress-test with will-it-scale: page_fault and mmap
Evaluation with will-it-scale

- **page_fault**
  - Graph 1: Comparison of ops/us between stock and BRAVO with varying # threads.
  - Description: Page fault performance comparison.

- **mmap**
  - Graph 2: Comparison of ops/us between stock and BRAVO with varying # threads.
  - Description: mmap performance comparison.

**Intel Xeon E7-8895 v3 CPU**
- 4 sockets
- 144 logical CPUs in total
Conclusion

• Builds into any existing lock
• Reads are accelerated
• Avoids write overhead
• Very compact
• Overall, takes the “reader indicator” dilemma away

Thank you! Questions?
Future Work

• Dynamic table sizing

• Probing multiple table locations

• Adaptive policies for enabling bias

• Revocation scan via SIMD instructions and non-temporal loads