IronFleet: Proving Practical Distributed Systems Correct

Hawblitzel et al.

Scriber: Haozhen Ding
Recap

IronFleet

- **Provable correctness of safety and liveness of distributed system implementation**

Methodology

- **Two-layer refinement**
Recap

Methodology
- Floyd-Hoare verification (Dafny, Z3)
- Temporal Logic of Actions (TLA) (for liveness)

Techniques
- Always-enabled actions (for liveness)
- Concurrency containment via reduction
- Invariant quantifier hiding (constructive proof)
- etc.

Implementation/Evaluation
- IronRSL (replicated state-machine library)
- IronKV (sharded key-store)
### Pros

+ Formal guarantees
+ Both safety and liveness
+ Novelty in two-layer refinement
+ Two verified systems have comparable performance
+ Near-real-time IDE feedback
+ Libraries
+ Lesson learned section
+ Fair assumptions
  + Non-reliable network

### Cons

- Much development effort
  - Proof code = 8x impl. Code
  - 3.7 person-years
- SMT solver complexity, need hints
- Dafny (or something similar)
- Compatibility with C++, Java?
- Hardness of heap management
- Exp. programs are CPU-bound
- Single threaded impl. on each host
- Formal proof of the atomicity reduction argument is future work
Discussion Questions

- IronFleet requires up to 8x lines of code for proof in addition to code yet achieves average performance. How do we balance the tradeoff between performance optimization and formal guarantee? **Is it worth the effort?**
<table>
<thead>
<tr>
<th></th>
<th>Spec (source lines of code)</th>
<th>Impl</th>
<th>Proof</th>
<th>Time to Verify (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High-Level Spec:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IronRSL</td>
<td>85</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>IronKV</td>
<td>34</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Temporal Logic</td>
<td>208</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>Distributed Protocol:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IronRSL Protocol</td>
<td>–</td>
<td>–</td>
<td>1202</td>
<td>4</td>
</tr>
<tr>
<td>Refinement</td>
<td>35</td>
<td>–</td>
<td>3379</td>
<td>26</td>
</tr>
<tr>
<td>Liveness</td>
<td>167</td>
<td>–</td>
<td>7869</td>
<td>115</td>
</tr>
<tr>
<td>IronKV Protocol</td>
<td>–</td>
<td>–</td>
<td>726</td>
<td>2</td>
</tr>
<tr>
<td>Refinement</td>
<td>36</td>
<td>–</td>
<td>3998</td>
<td>12</td>
</tr>
<tr>
<td>Liveness</td>
<td>98</td>
<td>–</td>
<td>2093</td>
<td>23</td>
</tr>
<tr>
<td>TLA Library</td>
<td>–</td>
<td>–</td>
<td>1824</td>
<td>2</td>
</tr>
<tr>
<td><strong>Implementation:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IO/Native Interface</td>
<td>591</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Common Libraries</td>
<td>134</td>
<td>833</td>
<td>7690</td>
<td>13</td>
</tr>
<tr>
<td>IronRSL</td>
<td>6</td>
<td>2941</td>
<td>7535</td>
<td>152</td>
</tr>
<tr>
<td>IronKV</td>
<td>6</td>
<td>1340</td>
<td>2937</td>
<td>42</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>1400</td>
<td>5114</td>
<td>39253</td>
<td>395</td>
</tr>
</tbody>
</table>

**Figure 12.** Code sizes and verification times.
Figure 14. IronKV’s performance is competitive with Redis, an unverified key-value store. Results averaged over 3 trials.
Discussion Questions

• IronFleet requires up to 8x lines of code for proof in additional to code yet achieves average performance. How do we balance the tradeoff between performance optimization and formal guarantee? **Is it worth the effort?**

System requirement
• Consistency vs availability
• Failure recovery

Business concern
Discussion Questions

- What are still in the protocol / implementation models assumed in IronFleet?
  - File storage?
  - Multi-threaded program?
  - Failure recovery?
Discussion Questions

- What are still missing in the protocol / implementation models assumed in IronFleet?
  - File storage? (memory)
  - Multi-threaded program? (not clear, additional proof)
  - Failure recovery? (part of distributed protocol)
Discussion Questions

- The paper proves Paxos liveness based on bounded message delay while in real network Paxos is not live. It might be that IronFleet verifies the correctness of a system but it is actually built upon unrealistic assumptions. How much can we trust our assumptions or the result of IronFleet?
Discussion Questions

- The paper proves Paxos liveness based on bounded message delay while in real network Paxos is not live. It might be that IronFleet verifies the correctness of a system but it is actually built upon unrealistic assumptions. How much can we trust our assumptions or the result of IronFleet?

At least as much as we can trust them without verification.
Discussion Questions

- The paper proves Paxos liveness based on bounded message delay while in real network Paxos is not live. It might be that IronFleet verifies the correctness of a system but it is actually built upon unrealistic assumptions. How much can we trust our assumptions or the result of IronFleet?

  At least as much as we can trust them *without* verification.

- Is it bad to assume the correctness of hardware, OS, compilers, Dafny, etc?
Discussion Questions

- The paper proves Paxos liveness based on bounded message delay while in real network Paxos is not live. It might be that IronFleet verifies the correctness of a system but it is actually built upon unrealistic assumptions. How much can we trust our assumptions or the result of IronFleet?

At least as much as we can trust them without verification.

- Is it bad to assume the correctness of hardware, OS, compilers, Dafny, etc?

No. We need layers of abstraction.
Discussion Questions

- The entire IronFleet suit took 3.7 human-years to build. Can we cut the development time in the future?
Discussion Questions

- The entire IronFleet suit took 3.7 human-years to build. Can we cut the development time in the future?

Certainly

- More verified common libraries
- Lessons learned about proof techniques
- Incremental change to codebase may not need more proofs
- Verification-aware development community
Discussion Questions

- Piazza: How comparable is IronFleet to Maude (from UIUC)?