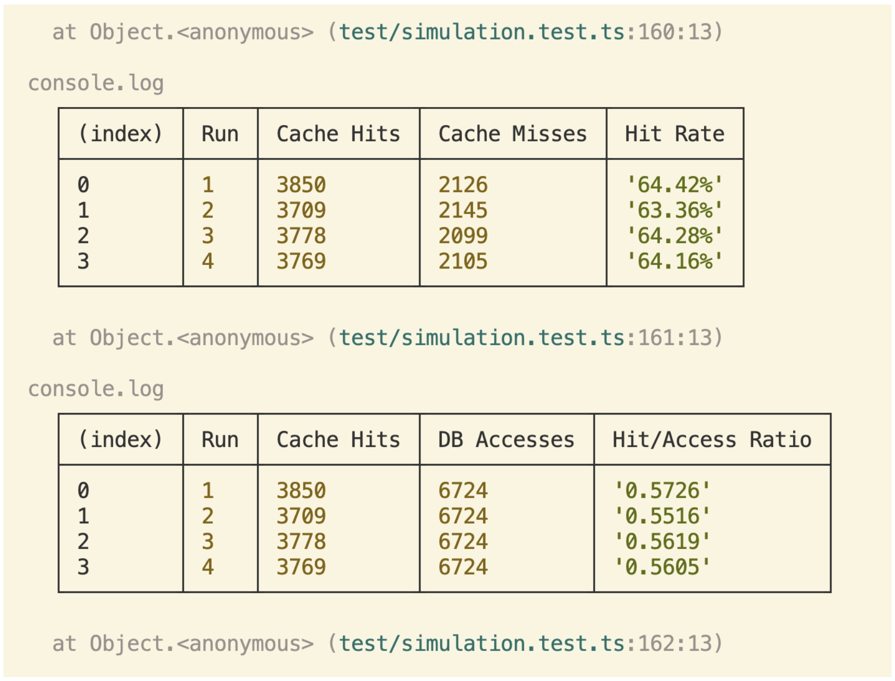
CS253P Project Report

Alexander Sanna

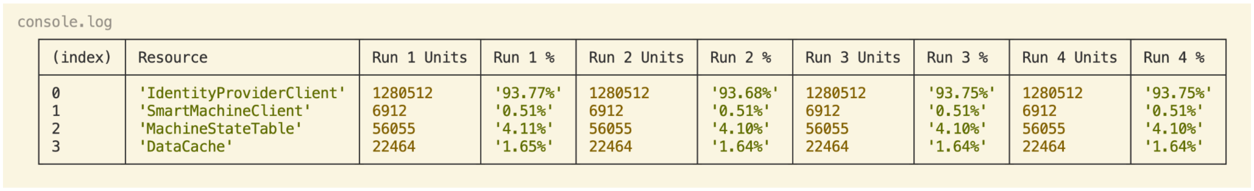
November 29th, 2025

Hamprr Service Analysis



```
  at Object.<anonymous> (test/simulation.test.ts:160:13)

console.log
  ┌─────────┬─────┬────────────┬──────────────┬──────────┐
  │ (index) │ Run │ Cache Hits │ Cache Misses │ Hit Rate │
  ├─────────┼─────┼────────────┼──────────────┼──────────┤
  │ 0       │ 1   │ 3850       │ 2126         │ '64.42%' │
  │ 1       │ 2   │ 3709       │ 2145         │ '63.36%' │
  │ 2       │ 3   │ 3778       │ 2099         │ '64.28%' │
  │ 3       │ 4   │ 3769       │ 2105         │ '64.16%' │
  └─────────┴─────┴────────────┴──────────────┴──────────┘

  at Object.<anonymous> (test/simulation.test.ts:161:13)

console.log
  ┌─────────┬─────┬────────────┬─────────────┬──────────────────┐
  │ (index) │ Run │ Cache Hits │ DB Accesses │ Hit/Access Ratio │
  ├─────────┼─────┼────────────┼─────────────┼──────────────────┤
  │ 0       │ 1   │ 3850       │ 6724        │ '0.5726'         │
  │ 1       │ 2   │ 3709       │ 6724        │ '0.5516'         │
  │ 2       │ 3   │ 3778       │ 6724        │ '0.5619'         │
  │ 3       │ 4   │ 3769       │ 6724        │ '0.5605'         │
  └─────────┴─────┴────────────┴─────────────┴──────────────────┘

  at Object.<anonymous> (test/simulation.test.ts:162:13)
```

All of my test cases have passed without conflicts. From my testing scripts I'm seeing on average 64% cache hits on test one and a 55% cache hits on test two. Meaning that a little more than half of the time we are able to access the data we need from a high speed cache - and the other times we are forced to take more time by going to the database.



```
console.log
  ┌─────────┬─────────────────────────┬─────────────┬──────────┬─────────────┬──────────┬─────────────┬──────────┬─────────────┬──────────┐
  │ (index) │ Resource                │ Run 1 Units │ Run 1 %  │ Run 2 Units │ Run 2 %  │ Run 3 Units │ Run 3 %  │ Run 4 Units │ Run 4 %  │
  ├─────────┼─────────────────────────┼─────────────┼──────────┼─────────────┼──────────┼─────────────┼──────────┼─────────────┼──────────┤
  │ 0       │ 'IdentityProviderClient'│ 1280512     │ '93.77%' │ 1280512     │ '93.68%' │ 1280512     │ '93.75%' │ 1280512     │ '93.75%' │
  │ 1       │ 'SmartMachineClient'    │ 6912        │ '0.51%'  │ 6912        │ '0.51%'  │ 6912        │ '0.51%'  │ 6912        │ '0.51%'  │
  │ 2       │ 'MachineStateTable'     │ 56055       │ '4.11%'  │ 56055       │ '4.10%'  │ 56055       │ '4.10%'  │ 56055       │ '4.10%'  │
  │ 3       │ 'DataCache'             │ 22464       │ '1.65%'  │ 22464       │ '1.64%'  │ 22464       │ '1.64%'  │ 22464       │ '1.64%'  │
  └─────────┴─────────────────────────┴─────────────┴──────────┴─────────────┴──────────┴─────────────┴──────────┴─────────────┴──────────┘
```

The important thing to note from this test is the consistency. We have clear set in stone values we are getting for each test, showing what resource is taking the most computational power. Happy to see that our data cache is taking very little resource.