

253P Report - System design

Hitha Shri Nagaruru

```
hithashrinagaruru@Mac cs-253p-hw-hampr-machine-service % npm test
hithashrinagaruru@Mac cs-253p-hw-hampr-machine-service % npm test
> hampr-base@1.0.0 test
> jest

PASS  test/api.test.ts
PASS  test/simulation.test.ts
● Console
  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' | 3840256    | '50.83%' | 3840256    | '50.76%' | 3840256    | '50.84%' | 3840256    | '50.76%'
    1     | 'SmartMachineClient'   | 870976     | '11.53%' | 870976     | '11.51%' | 870976     | '11.53%' | 870976     | '11.51%'
    2     | 'MachineStateTable'   | 2030022    | '37.74%' | 2030022    | '37.73%' | 2030022    | '37.74%' | 2030022    | '37.73%'
    3     | 'DataCache'            | 46182      | '0.61%'  | 46182      | '0.61%'  | 46182      | '0.61%'  | 46182      | '0.61%'

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

    (index) | Run | Cache Hits | Cache Misses | Hit Rate
    0     | 1   | 1898       | 1134        | '62.60%' 
    1     | 2   | 1879       | 1120        | '62.51%' 
    2     | 3   | 1935       | 1126        | '63.21%' 
    3     | 4   | 1830       | 1109        | '62.27%' 

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

    (index) | Run | Cache Hits | DB Accesses | Hit/Access Ratio
    0     | 1   | 1898       | 23871       | '0.0795' 
    1     | 2   | 1874       | 23871       | '0.0785' 
    2     | 3   | 1935       | 23871       | '0.0811' 
    3     | 4   | 1830       | 23871       | '0.0767' 

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

Test Suites: 2 passed, 2 total
Tests:       12 passed, 12 total
Snapshots:  0 total
Time:        1.065 s
Ran all test suites.
hithashrinagaruru@Mac cs-253p-hw-hampr-machine-service %
```

The performance statistics show that the service behaves consistently across multiple runs and that most of the computational load is dominated by the IdentityProviderClient and MachineStateTable, which together account for roughly 88% of total execution units. The SmartMachineClient contributes a much smaller portion, and the in-memory DataCache has almost negligible overhead, which is expected since caching is lightweight. The cache hit rate stays stable around 62–63%, meaning the cache is doing useful work but there's still room for improvement if we want to reduce database accesses further. Even with that moderate hit rate, the DB is still accessed heavily (around 23k accesses per run), which aligns with the simulation's design of repeatedly pulling machine state. Overall, the system performs reliably, the workload distribution matches expectations for this architecture, and all tests pass cleanly with consistent timing across runs.