LF NETWORKING
Developer & Testing Forum

CPS Updates & Demo

Toine Siebelink

May 2024

ERICSSON ONAP CPS

https://lfnetworking.org

Who am I ?

Toine Siebelink

- Master Engineer with Ericsson 20+ Years
- Working in Open Source (ONAP) since 2020 as Project Technical Lead for CPS
- My Focus
  - Quality Testware
  - Quality Software
  - Performance
- CPS has achieved OpenSSF Gold Badge Best Practices (quality & security)

ERICSSON

CPS

Agenda

1. Quick Recap of CPS Concepts
2. Demo of Delta Feature
3. Performance Improvements & Micro Benchmark Testing
4. Warm Up Effect
5. Performance Monitoring

LF NETWORKING

LFN Developer & Testing Forum

CPS Concepts

Dataspace, Model Sets, Anchors

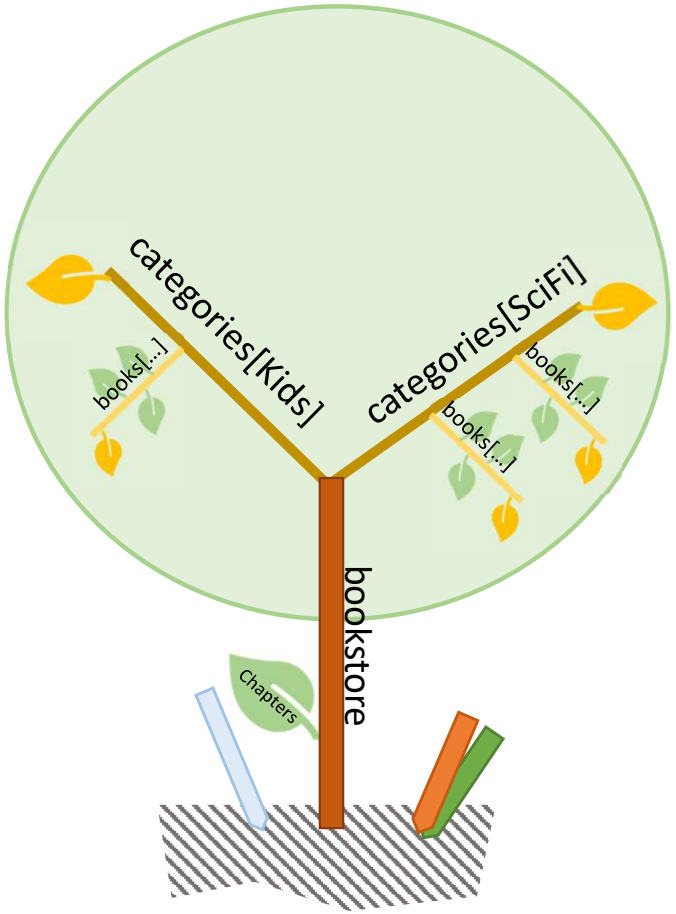
ERICSSON

CPS

Concepts: Model, Data & Instance Tree

Model.yang

```
module stores {
    yang-version 1.1;
    namespace "org:onap:ccsdk:sample";

    prefix book-store;

    revision "2020-09-15" {
        description
        "Sample Model";
    }

    typedef year {
        type uint16 {
            range "1000..9999";
        }
    }

    container bookstore {

        leaf bookstore-name {
            type string;
        }

    list categories {

        key "name";

        leaf name {
            type string;
        }

        list books {
```



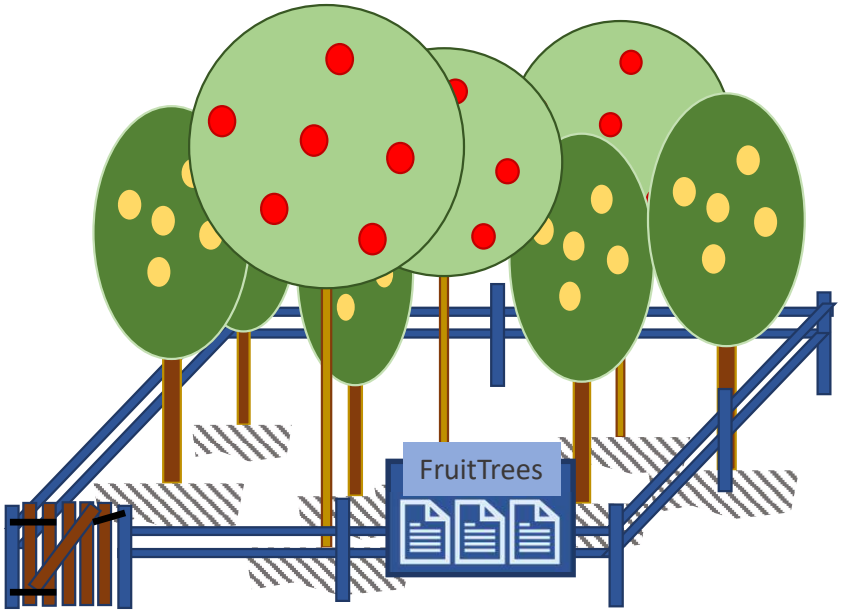
Data.json


```
{
  "test:bookstore":{
    "bookstore-name": "Chapters",
    "categories": [
      {
        "code": "01",
        "name": "SciFi",
        "books": [
          {
            "author": "Iain M. Banks",
            "lang": "en",
            "price": "895",
            "pub_year": "1994",
            "title": "Feersum Endjinn"
          },
          {
            "author": "Dan Simmons",
            "lang": "en",
            "price": "1099",
            "pub_year": "1999",
            "title": "Far Horizons"
          }
        ]
      },
      {
        "name": "kids",
        "code": "02",
        "books": [
          {
            "author": "Philip Pullman",
            "lang": "en",
```


ERICSSON

CPS

Concepts: Dataspaces



- A ‘dataspace’ is like the fenced in patch of land containing only certain ‘types’ of trees.
- A dataspace contains a list of the types of trees it can contain i.e. the Yang modules that describe the data trees
- An application defines a dataspace (name) and is responsible for maintaining the models in it.

- Each dataspace has an owner who controls the gate to determine who can access the trees.

ERICSSON

CPS

LF NETWORKING

LFN Developer & Testing Forum

Live Demo 1

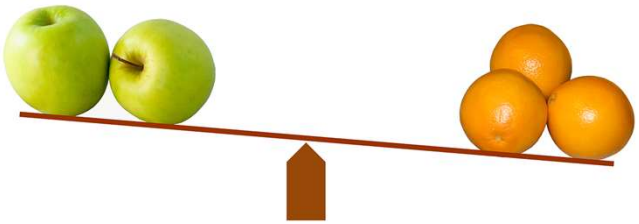
Delta Feature

ERICSSON

CPS

Delta Feature

- Community initiated contribution
  - Compare data for different anchors



- Context:
  - Storing CM snapshots over time
  - Planned configuration changes

- Use Cases
  - Compare planned changes with recent snapshots
  - See impact of proposed configuration plans
  - Compare live configurations
  - Delta notifications (future)

- Can compare ‘Book Stores’ too 😄

Delta Feature

- Output
  - Based on IETF standards
    - RFC6902: JSON Patch
    - RFC9144 Comparison of Network Management Datastore Architecture (NMDA) Datastores
  - ADD
  - REMOVE
  - UPDATE

- Deutsche Telecom are planning to use CPS Delta Feature in production for next year

Delta Feature Demo

- Swagger-UI
- Create
  - Dataspace
  - Model Sets
  - Anchors
  - Data
- Query
- Delta Report
  - Update
  - Add
  - Remove

LF NETWORKING

LFN Developer & Testing Forum

Performance

Improvements & Micro Benchmark Testing

ERICSSON

CPS

Performance Improvements

Slogan	Effect	Test
Enable Hibernate Batching	> 2x (store operations)	`org.onap.cps.integration.performance.cps.WritePerfTest# Writing openroadm data has linear time.`
Normalize JSON for store & update	~ 2x (update operations)	`org.onap.cps.integration.performance.cps.UpdatePerfTest #Replace single data node and descendants: #scenario.`
Saving CM Handles (NCMP Network Configuration Mgmt. & Persistence) Batch yang data parsing	4x	
Faster absolute cps-path queries	5-10x (CPS Queries)	`org.onap.cps.integration.performance.ncmp.CmHandleQuery PerfTest#CM-handle is looked up by alternate-id.`
Removed Redundant Spring Security (using service mesh instead)	Overhead 100ms -> 10ms	

Micro-Benchmark Testing

- Spock & Groovy ‘unit’ test Framework
  - Given
  - When
  - Then
  - Where
- Spring-boot Test & Test Containers
  - PostgreSQLContainer
  - KafkaTestContainer
- Single Method, large data size and/or repeats
- Semi Integration
- Run fast & often in IDE

Micro Benchmark Test Demo

Base Test Class

```
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.MOCK, classes = [CpsDataspaceService])
@Testcontainers
@EnableAutoConfiguration
@AutoConfigureMockMvc
@EnableJpaRepositories(basePackageClasses = [DataspaceRepository])
@ComponentScan(basePackages = ['org.onap.cps'])
@EntityScan('org.onap.cps.spi.entities')
abstract class CpsIntegrationSpecBase extends Specification {

    @Shared
    DatabaseTestContainer databaseTestContainer = DatabaseTestContainer.getInstance()

    @Shared
    KafkaTestContainer kafkaTestContainer = KafkaTestContainer.getInstance()

    @Autowired
    MockMvc mvc

    @Autowired
    CpsDataspaceService cpsDataspaceService
```

Micro Benchmark Test Demo

Test Class

```
def 'Writing openroadm data has linear time.'() {
    given: 'an empty anchor exists for openroadm'
        cpsAnchorService.createAnchor(CPS_PERFORMANCE_TEST_DATASPACE, LARGE_SCHEMA_SET, WRITE_TEST_ANCHOR)
    and: 'a list of device nodes to add'
        def jsonData = generateOpenRoadData(totalNodes)
    when: 'device nodes are added'
        resourceMeter.start()
        cpsDataService.saveData(CPS_PERFORMANCE_TEST_DATASPACE, WRITE_TEST_ANCHOR, jsonData, OffsetDateTime.now())
        resourceMeter.stop()
    then: 'the operation takes less than #expectedDuration and memory used is within limit'
        recordAndAssertResourceUsage("Writing ${totalNodes} devices", expectedDuration, resourceMeter.getTotalTimeInSeconds()
    cleanup:
        cpsAnchorService.deleteAnchor(CPS_PERFORMANCE_TEST_DATASPACE, WRITE_TEST_ANCHOR)
    where:
        totalNodes || expectedDuration
        50         || 1
        100        || 2
        200        || 3
        400        || 5
}
```

LFN Developer & Testing Forum

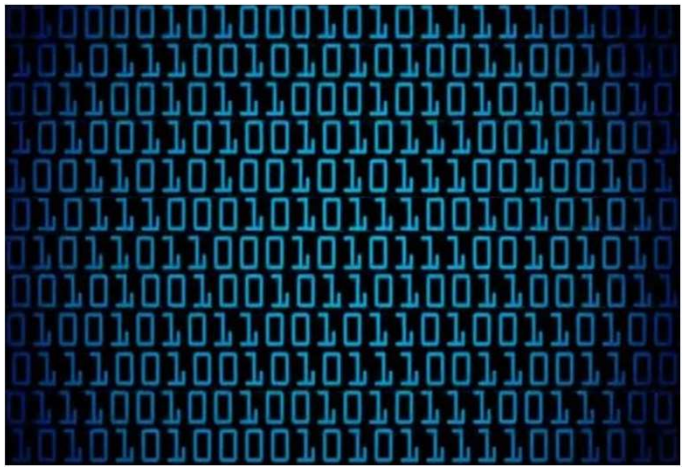
Live Demo 2

Micro Benchmark Testing

ERICSSON

CPS

Performance Improvements



1. Writing openroadm data has linear time
2. Replace single data node and descendants
3. CM-handle is looked up by alternate-id

LF NETWORKING

LFN Developer & Testing Forum

Warm Up Effect

ERICSSON

CPS

Warm Up Effect

- JIT: Just-In-Time Compilation
  - Lazy Class Loading, first call slower
  - Native Code -Cache
- Tiered Compilation in JVM (default since Java 8)
  - Interpreted with Profiling
  - C1 (client) Compiler with Profiling
  - C2 (server) Compiler, non-profiled

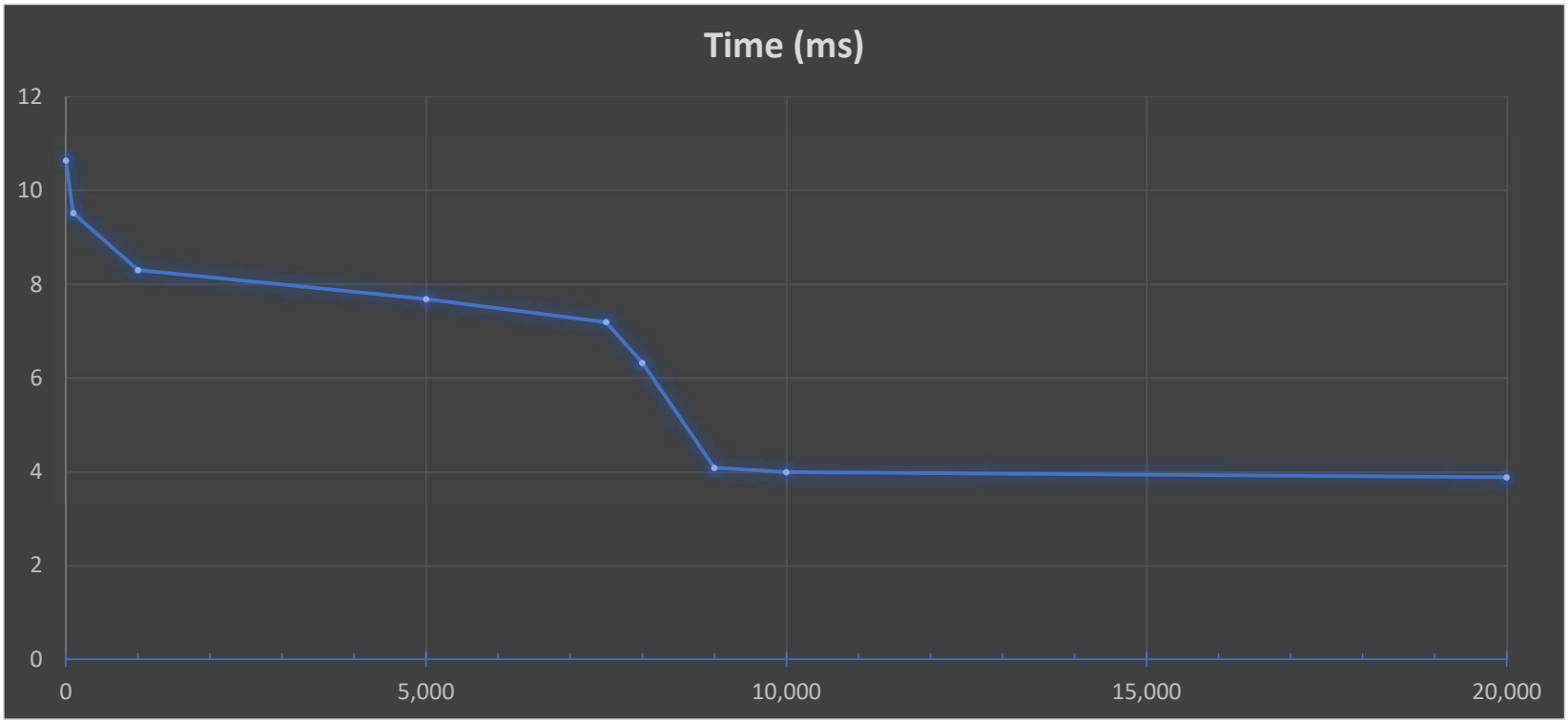
References

- https://www.baeldung.com/java-jvm-warmup
- https://www.baeldung.com/jvm-tiered-compilation

Warm Up Effect

LFN Developer & Testing Forum

Effect of Warm Up on CM-Handle Look-Up Query (100 samples)



ERICSSON

CPS

LF NETWORKING

LFN Developer & Testing Forum

Performance Monitoring

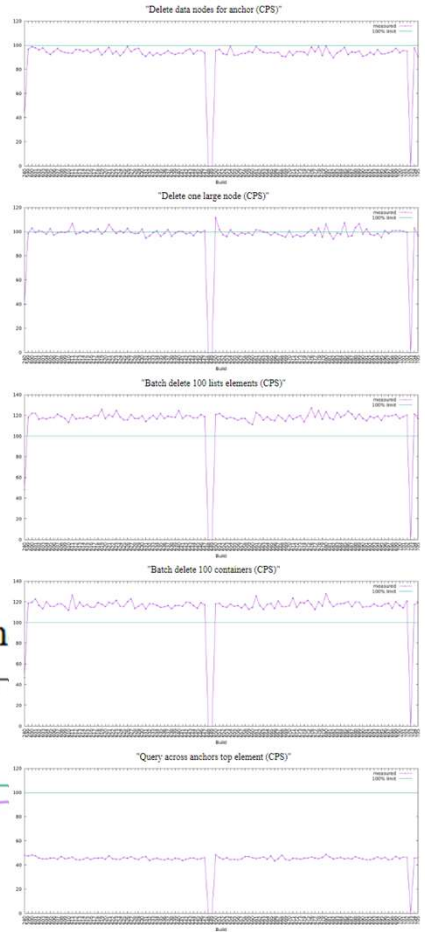
Prevent Performance Regression

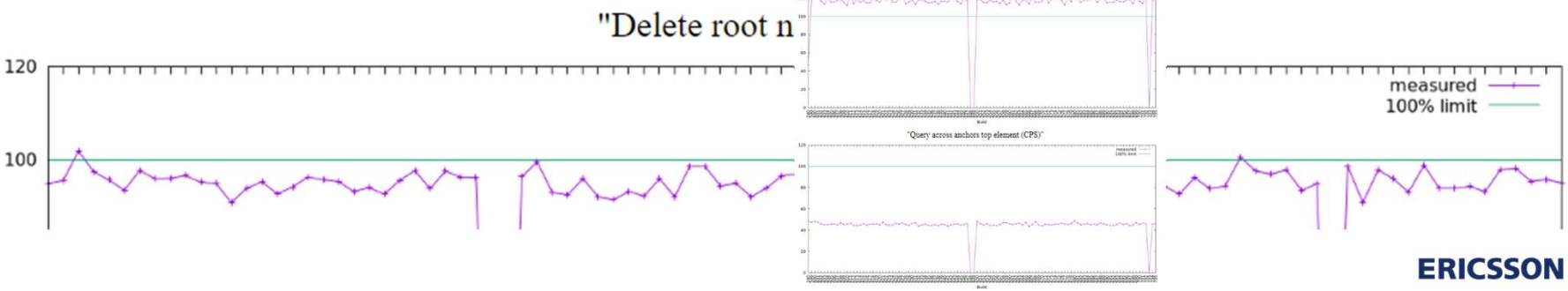
ERICSSON

CPS

Performance Monitoring

- Run micro benchmark tests regularly (2h)
- Jenkins plugin to store and graph results
- Assess graphs every day during Standup
- Look for degradation
- Confirm improvements when tuning/fixing performance issues



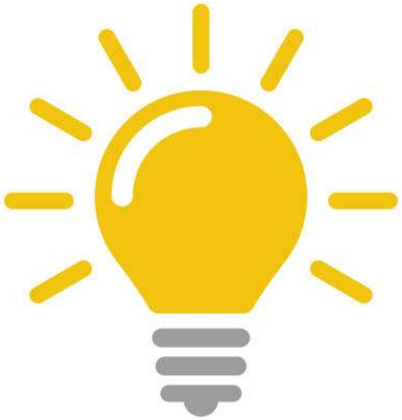


ERICSSON

CPS

Performance Monitoring

Lessons Learned

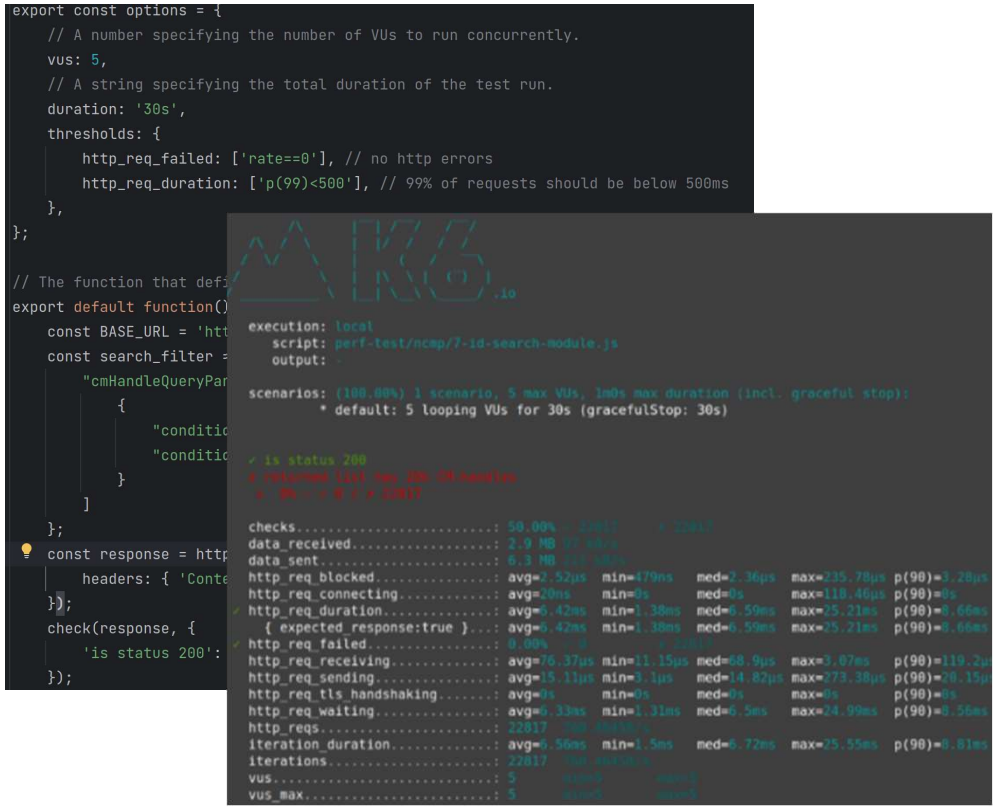


- Reasonable size test for less variation
- Run in well controlled environment (physical)
- Run multiple times day , not just nightly
- Linux / Windows, Laptop/Server Virtual/Physical affects different test differently
- Warm Up Effect

Future Test Improvements

- K6 (performance) Test
  - Full End 2 End
  - Using REST Interface
  - Designed for Performance Test
    - Concurrent users
  - More advance criteria
  - More control
    - Repeat requests in loop
    - Duration based instead of loop

- PoC in Progress



ERICSSON

CPS

LF NETWORKING

LFN Developer & Testing Forum

Stay in Touch

- Toine.Siebelink@est.tech
- https://wiki.onap.org/display/DW/Configuration+Persistence+Service+Developer%27s+Landing+Page

ERICSSON

CPS

References

- https://wiki.onap.org/display/DW/Configuration+Persistence+Service+Developer%27s+Landing+Page
- https://spockframework.org/
- https://www.baeldung.com/java-jvm-warmup
- https://www.baeldung.com/jvm-tiered-compilation
- https://plugins.jenkins.io/htmlpublisher/
- https://k6.io/
- https://www.bestpractices.dev/en/projects/4398
- ONE Summit 2022: Spock & Groovy