CERN

EOS Citrine Scheduler tutorial

Andrea Manzi

EOS Workshop 2019

4/5 February 2019

Overview

- **EOS Citrine Scheduler**
  - File Scheduling in EOS
  - Implementation: GeoTreeEngine
- **Demo**
  - FST geotag and Client geotag
  - GeoTreeEngine conf and state
  - Branch disabling
  - Placement policies
  - RAIN layouts

File Scheduling in EOS Citrine

- **File scheduling** is the process of deciding by which (FST) server a user request is to be served and it’s carried out in the **MGM** node

- EOS Citrine implementation (EOS 4.x)
  - **Infrastructure aware scheduling** supporting multiple locations and hierarchical nesting
  - Implement placement policies compatible with **all layouts**
  - Proxy selection for **non-native filesystems**
  - **Firewall entry point** selection

- In production @ CERN since ~2 years

- No changes in the last year, apart from bug fixes

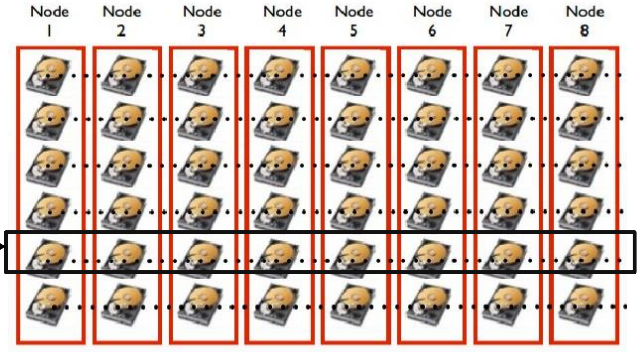
Scheduling Groups, Geotags and Trees

EOS space (simplified)

- Set of machines acting as storage servers

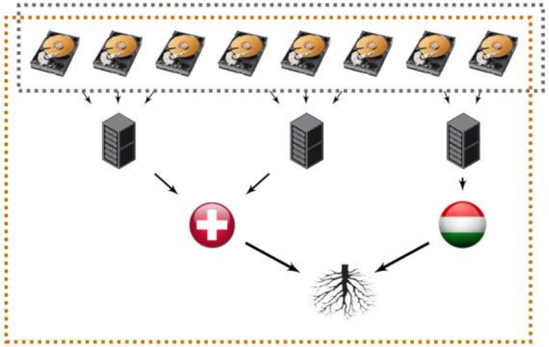
EOS scheduling group

- Set of filesystems (drives) scattered across distinct machines in a space

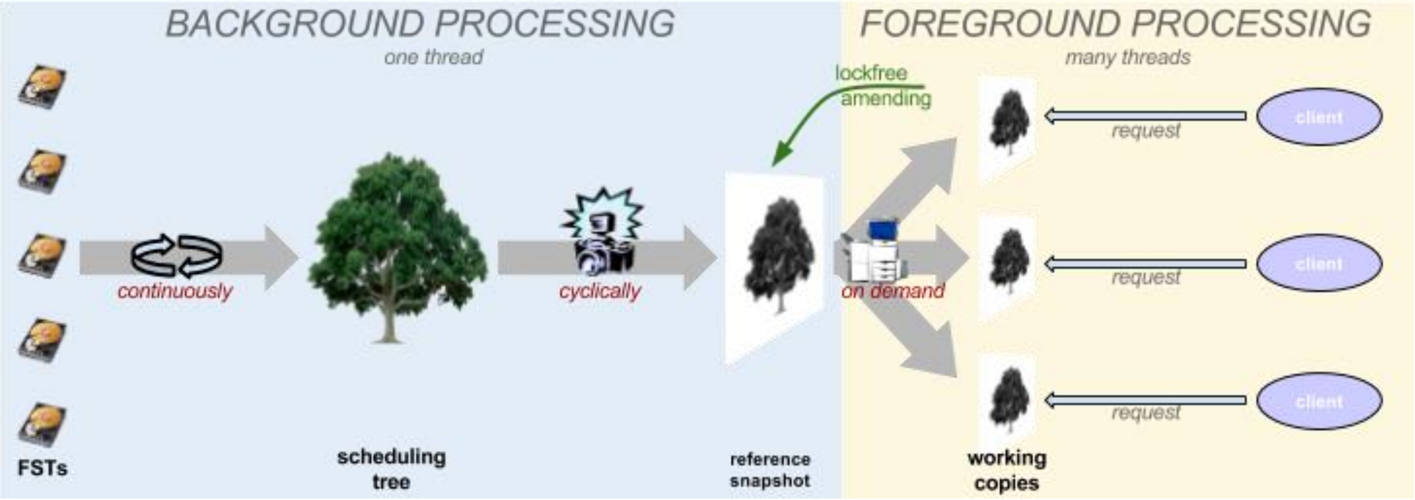


EOS Citrine scheduler

- Each filesystem inherit host's geotag featuring arbitrary depth e.g. CERN::513::RACK1
- The scheduling is performed **among** the filesystems part of a scheduling group seen with a **tree** structure.
  - For file placement the selection of the scheduling group to use is Round Robin



Implementation: The GeoTreeEngine



- **Trees, snapshots**: they contain information about the filesystems: status, free space, ul/dl score, free slots, taken slots
- **Background updater**: receive update notifications for the fs and keeps trees and reference snapshots up to date
- **On demand**: working copies of reference snapshots are used to place/access new file.
- **Latency estimation and penalty subsystem**: lock-free system to avoid overscheduling in bursts of requests
- **Implementation**: lock-free, memory efficient, scalable, low latency

Demo - geotags

- FST **Geotag** configuration
  - /etc/sysconfig/eos (/etc/sysconfig/eos_env)
    - (export) EOS_GEOTAG=“CERN::513::RACK1“
    - N.B. 8 chars limitation on each geotag portion (to be enforced by FSTs from the next version)
    - N.B. Since EOS 4.1.x scheduling will work even without EOS_GEOTAG configured

- Set **client** geotag
  - Clients geotags are attributed by the MGM using rules
  - vid set geotag <IP-prefix> <geotag>
  - When placing/accessing files, 1 replica will be stored/accessed closest to the client geotag
  - N.B. Properly working since EOS v4.2.23

Demo – GeoTreeEngine configuration

- **GeoTreeEngine** Configuration **:**
  - Show the config with the command

```
geosched show param
```

  - Alter the config with the command

```
geosched set <param name> [param index] <param value>
```

  Some parameters : skipSaturatedPlct, skipSaturatedAccess, fillRatioLimit, fillRatioCompTol, saturationThres

- Check **state** of the engine and tree/snapshots
  - Show

```
geosched show [tree|snapshot|state]
```

- **Disable subtrees** for selected operations

```
geosched disabled [add|rm|show] <geotag> <optype> <group>
```

Demo - Placement Policies

- It is a **scheduling information**, **NOT** a file property or attribute

	gathered:*tag1::tag2*	**hybrid:*tag1::tag2***	**scattered:tag1::tag2 (default)**
Replica	all as close as possible to *tag1::tag2*	all-1 around *tag1::tag2* 1 as scattered as possible	all as scattered as possible
RAIN	all as close as possible to *tag1::tag2*	all-n_parity around *tag1::tag2* n_parity as scattered as possible	all as scattered as possible

- Specify placement policies **in multiple contexts**
  - Set placement policy in a directory

```
eos attr set sys.forced.placementpolicy=gathered:site2 /eos/demo
```

  - Specify placement policy in an explicit file conversion

```
eos file convert /eos/demo/passwd replica:2 default scattered
```

  - Set placement policy in an automatic conversion (LRU converter)

```
eos attr set 'sys.conversion.*=00600112|scattered' /eos/demo
```

Demo – Scheduler with RAIN layouts

- EOS supports 3 types of RAIN layouts:

	redundancy	algorithm	description
raiddp	4 + 2	Dual parity raid	can lose 2 disks without data loss
raid6	N + 2	Erasure Code (Jerasure library)	can lose 2 disks without data loss
archive	N + 3	Erasure Code (Jerasure library)	can lose 3 disks without data loss

- **Per directory layout:**
  - attr set default=archive /eos/instance/archive
  - attr set sys.forced.stripes=10 /eos/instance/archive

Questions?