
CERN

EOS Citrine Scheduler and Centralized Drain

Geoffray Adde

Andrea Manzi

EOS Workshop 2018

5/6 February 2018

Overview

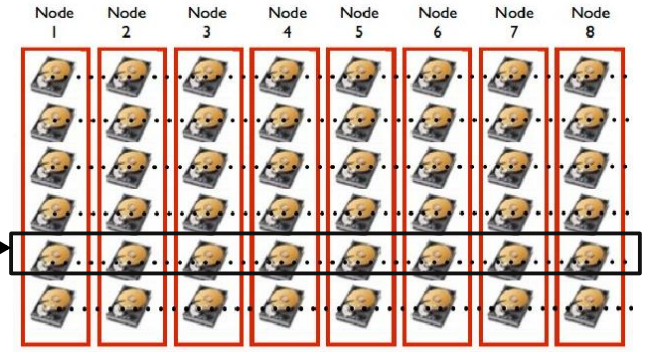
- **EOS Citrine Scheduler**
  - File Scheduling in EOS
  - Tree-based Selection Algorithm
  - Implementation: GeoTreeEngine
  - Configuration & CLI
  - Data Proxies, Proxy Groups and Firewall Entrypoints
- **EOS Citrine Centralized Drain**
  - How it works
  - Configuration & CLI
- **Next steps**

File Scheduling in EOS Citrine

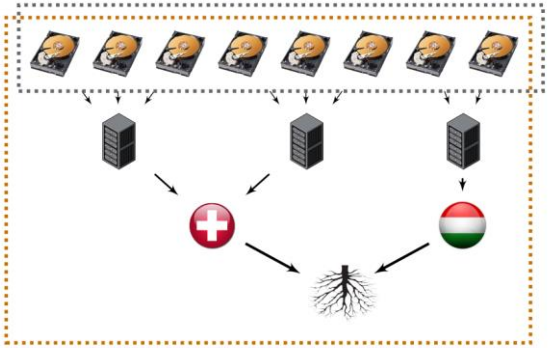
- **File scheduling** is the process of deciding by which (FST) server a user request is to be served
- It is carried out in the **MGM** node
- EOS Citrine implementation (EOS 4.x)
  - **Infrastructure aware scheduling** supporting multiple locations and hierarchical nesting
  - Implement placement policies compatible with **all layouts**
  - Proxy selection for **non-native filesystems** (e.g. Kinetic drives)
  - **Firewall entry point** selection
- In production @ CERN on all LHC VOs instances + Public instance

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

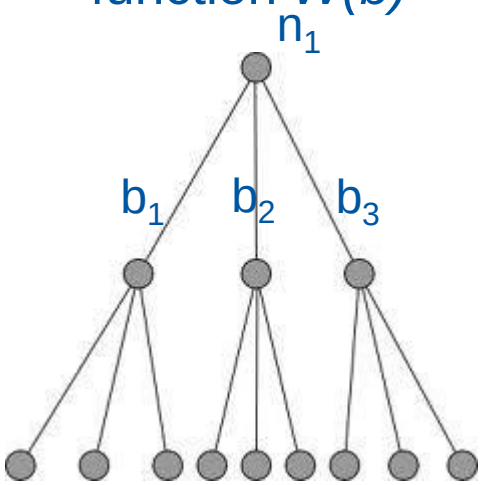


- **EOS Citrine scheduler**
- Each filesystem inherit host's geotag featuring arbitrary depth e.g. CC1::ROOM3::RACK2
- The scheduling is performed **among** the filesystems part of a scheduling group seen with a **tree** structure.

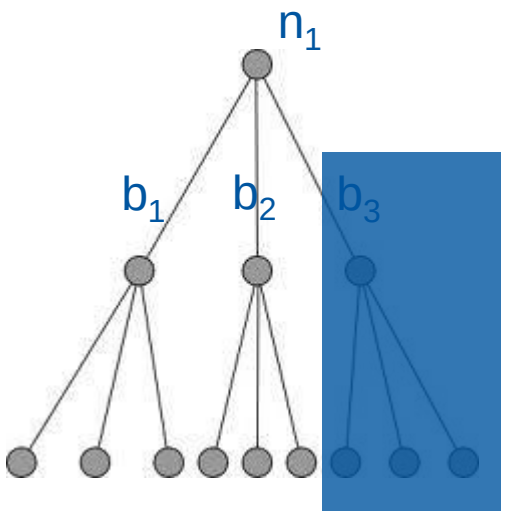


Tree-based Selection Algorithm

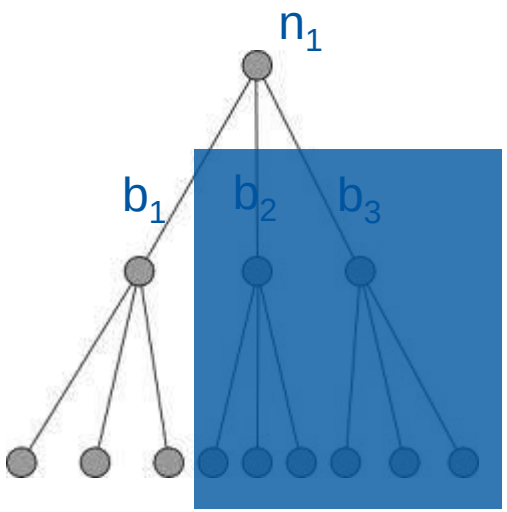
- **Purpose:** Select a free slot on a leaf
- **Rule:** Always go deeper in the tree, don't come back uproot
- *O(b,b')* is an order function that allows to compare branches *b* and *b'* at a given node
- When several best branches, randomly take using a sampling weight function *W(b)*



Stage 1 : Select Candidates
$O(b_1,b_2)=0$ and $O(b_2,b_3)>0$
Candidates are b_1 and b_2



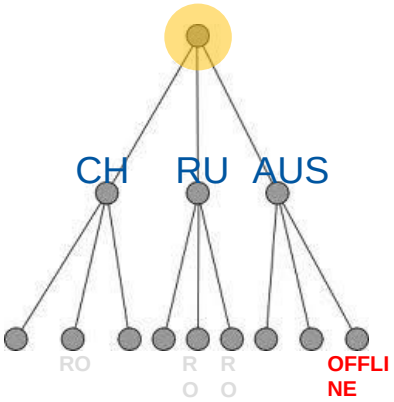
Stage 2 : Random Selection
$W(b_1)=40$ and $W(b_2)=60$
Selected candidate is b_1



Stage 3 : Go down the selected branch
And start again in Stage 1

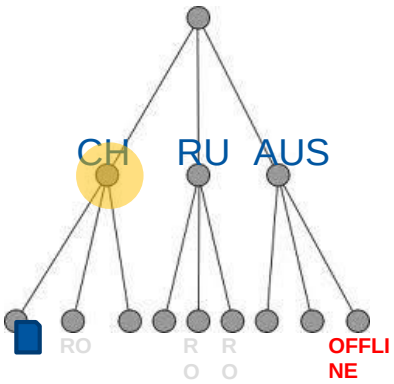
Selection algorithm: File Placement

- We consider a **scattered** file placement with 2 replicas
- Start the algorithm from the top (root) with:
  - Function *O* using availability, filling ratio, filling limit, number of replica.
  - Function *W* network IO, disk IO



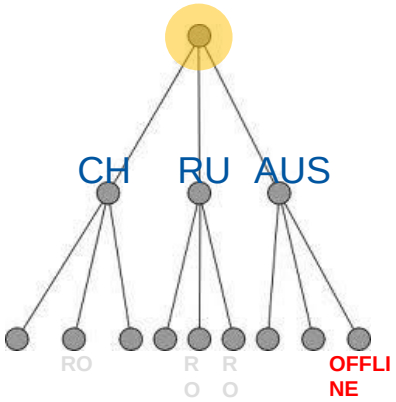
Step 1

- All branches equal
- Randomly choose CH



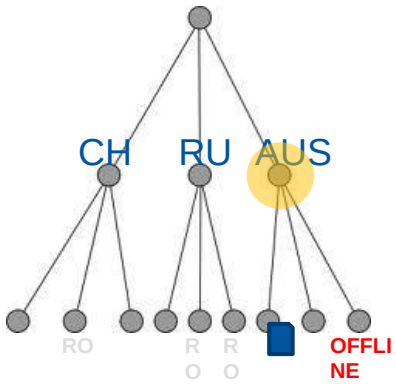
Step 2

- Branches 1=3 and 1>2 (because it is RO)
- 3 is very busy
- 1 is randomly selected (could have been 3 though)



Step 3

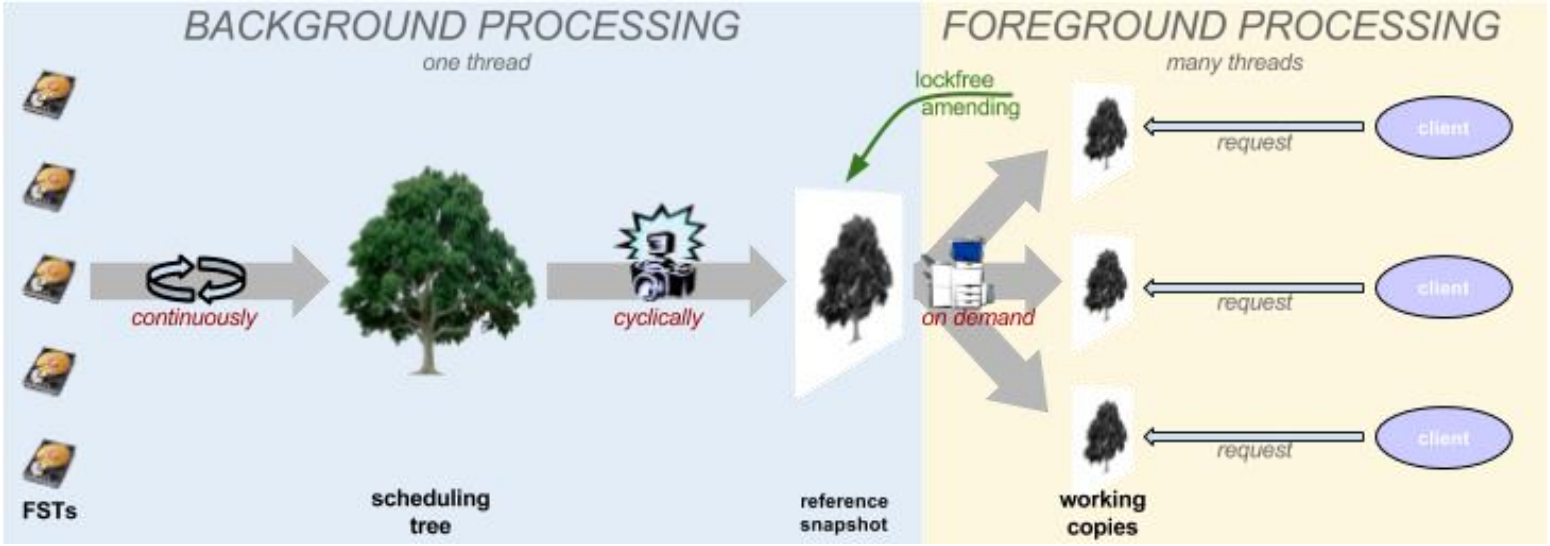
- Branches 2=3 and 1<2 (because 1 already has a replica)
- Randomly choose AUS



Step 4

- Branches 1=2 and 1>3 (because it is OFFLINE)
- 1 and 2 equally busy
- 1 is randomly selected (could have been 2 though)

Implementation: The GeoTreeEngine



- **Trees, snapshots**: they contain information about the filesystem needed to go deeper in the tree at each level: 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. Different snapshots for each operation (file placement, access, draining, balancing)
- **Latency estimation and penalty subsystem**: lock-free system to avoid overscheduling in bursts of requests

Scheduler Configuration (Admins)

- **http://eos-docs.web.cern.ch/eos-docs/configuration/geoscheduling.html**

- Set the **geotag** on each FST:
  - In /etc/sysconfig/eos (/etc/sysconfig/eos_env) add

    (export) EOS_GEOTAG="0513::R::0050::RJ03“
  - All the EOS filesystems hosted by the box will inherit this geotag
- Set **clients geotag** (in order to select the replica **closest** to client):
  - Clients geotags are attributed by the MGM using rules.

    vid set geotag <IP-prefix> <geotag>
- Set the geotag on **imported filesystems**:
  - On the MGM, use the command

    fs config <fsid> forcegeotag= 0513::R::0050::RJ03
  - This command also overrides the FST geotag for individual filesystems

Scheduler Configuration (Admins) [2]

- **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
- Manage **state** of the engine
  - Show

    geosched show [tree|snapshot|state]
  - Interact with the background updater

    geosched updater [pause|resume]

    geosched forcerefresh
- **Disable subtrees** for selected operations

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

Placement Policies (Users)

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

	gathered:*tag1::tag2*	**hybrid:*tag1::tag2***	**scattered**
Replica	all as close as possible to *tag1::tag2*	all-1 around *tag1::tag2* 1 as scattered as possible	all as scattered as possible
RAID	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

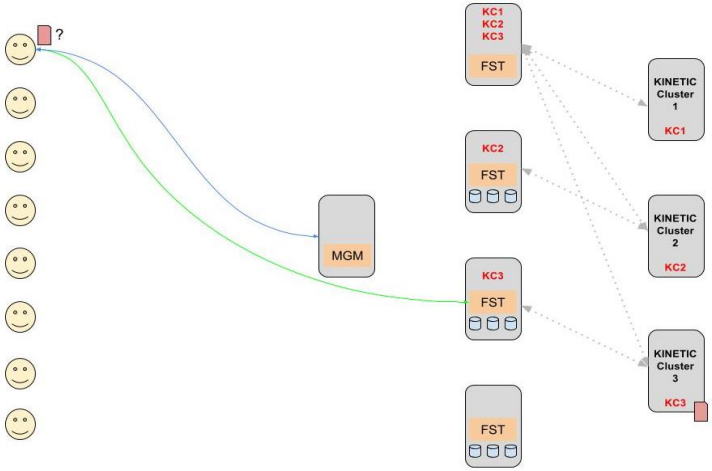
    eos attr set sys.forced.placementpolicy=**gathered:site2** /eos/demo
  - Specify placement policy in an explicit file conversion

    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

Data Proxy Scheduling

- Since EOS 4.1: imported storage Kinetic, plain Xrootd, RadosFS, http/S3
  - Imported storage accessed by FSTs named **Data Proxy**
  - For each imported storage an FST reports to the MGM the state
  - When scheduling imported storage, a Data Proxy must be scheduled too
  - Each filesystem can be associated to a **proxy group**, each FST can be part of multiple **proxy groups**



- **http://eos-docs.web.cern.ch/eos-docs/configuration/proxys.html**

Proxy Groups and Firewall Entrypoints

- Managing **Proxy Groups**
  - To manage proxy groups an FST is part of:

    node [proxygroupadd,proxygrouprm,proxygroupclear] …
    node status …
  - To manage the proxy group attached to a filesystem

    fs config <fsid> proxygroup=<somegroup>

    optional: fs config <fsid> filestickyproxydepth=<somegroup>

- A proxy group can be used to define **Firewall Entrypoints:**
  - Manage direct access (no firewall to cross)

    geosched access [setdirect|showdirect|cleardirect]
  - Map subtrees to proxy groups acting as entrypoints

    geosched access [setproxygroup|showproxygroup|clearproxygroup]

Centralized Drain

- New implementation of the filesystem drain (EOS 4.2.5)
- Makes use of the **GeoTreeEngine** in order to select the destination FS for draining. Runs and monitors **TPC** source FST => destination FST
- The current Drain implementation where each FST polls for replica to drain is not compatible with the placement policies implemented by GeoTreeEngine
- Tries to avoid **overloading** the system when large draining campaign are executed by limiting the number of FS to drain per FST

Centralized Drain[2]

- Possibility to select **a destination FS for draining**, if we just want to move replicas to a new FS
- **New CLI commands**
- Displays **drain errors** for a given drain activity
- FSs under drain are now set as **RO**, so clients can read data from them
- Does not replace the existing drain (yet)
  - Disabled by default

Configuration and CLI commands

- To **enable** the new drain in the MGM configuration
  - mgmofs.centraldraining true
- Space Configuration
  - **space.drainer.node.nfs**: max fs under drain per FST (default 5)
  - **space.drainer.fs.ntx**: max parallel drain transfers per fs (default 10)
  - **space.drainer.retries**: number of retries per drain process (default 1)
- New CLI Commands
  - **drain start/stop/status [fsid]**
  - **drain clear [fsid]** -> remove the drain status info for a given FS
- Compatible with existing commands
  - **fs ls -d** -> drain statistics
- Scheduler specific drain configuration
  - i.e. geosched disabled add CERN::0513 **plctdrain**

Next Steps

- GeoTreeEngine
  - Ability to configure custom **weights** per geotag, to be taken into account by the Tree-based selection algorithm
  - Use the GeoTreeEngine inside the various **balancers** (Infra/Inter Group, GEO)
- Drain
  - Implement drain for **RAIN** layout with automatic reconstruction (RAIN layout drain is not enabled for now)
- Analyze possible new requirements coming from **XDC project** & **WLCG Data Lake** initiative

Questions?