RTDS(LU-9809)
Real-Time Dynamic Striping

Li Xi

DataDirect Networks

Why RTDS(LU-9809)?

- **Current ways of controlling file striping are not enough**
  - Default striping
    - Only have a fixed policy based on free space
    - Not able to be controlled from outside
  - OST pool based striping
    - Can configure different stripes for different pools
    - Still not able to control the details of the policy
  - Create file with specific striping using flag O_LOV_DELAY_CREATE and ioctl(LL_IOC_LOV_SETSTRIPE)
    - Needs modification of the application
- **RTDS: a way to better control the striping**

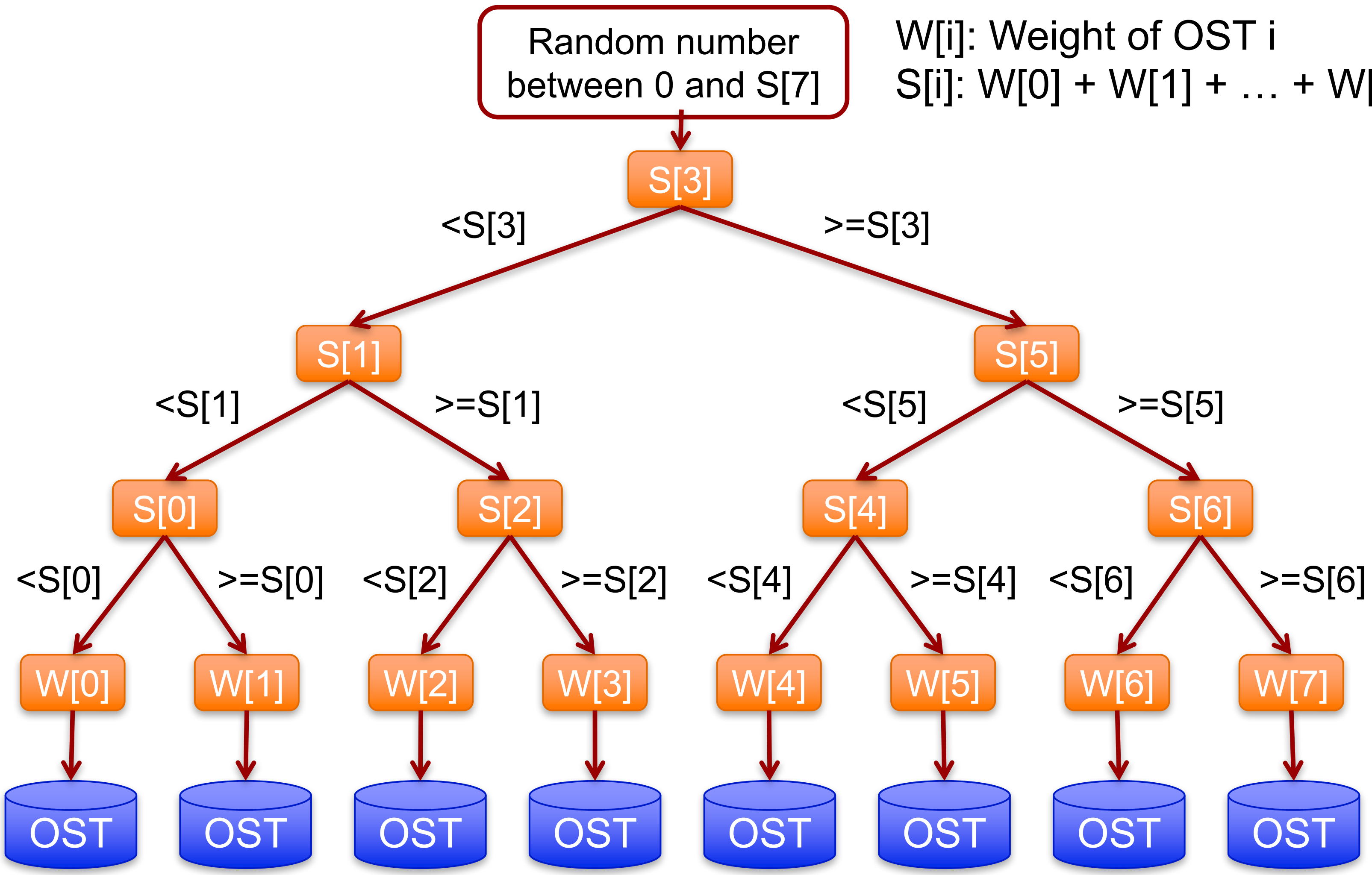
Design of RTDS(LU-9809)

- Each OST has a configurable weight
- When allocating an object, RTDS randomly choose an OST
- The probability of choosing a given OST is proportional to the OST's weight
- The administrator can configure the weights of all OSTs in real-time

```
# cat /proc/fs/lustre/lod/vm1-MDT0000-mdtlov/rtds_weight
0=1,1=1
# echo "0=1,1=2" > /proc/fs/lustre/lod/vm1-MDT0000-
  mdtlov/rtds_weight
```

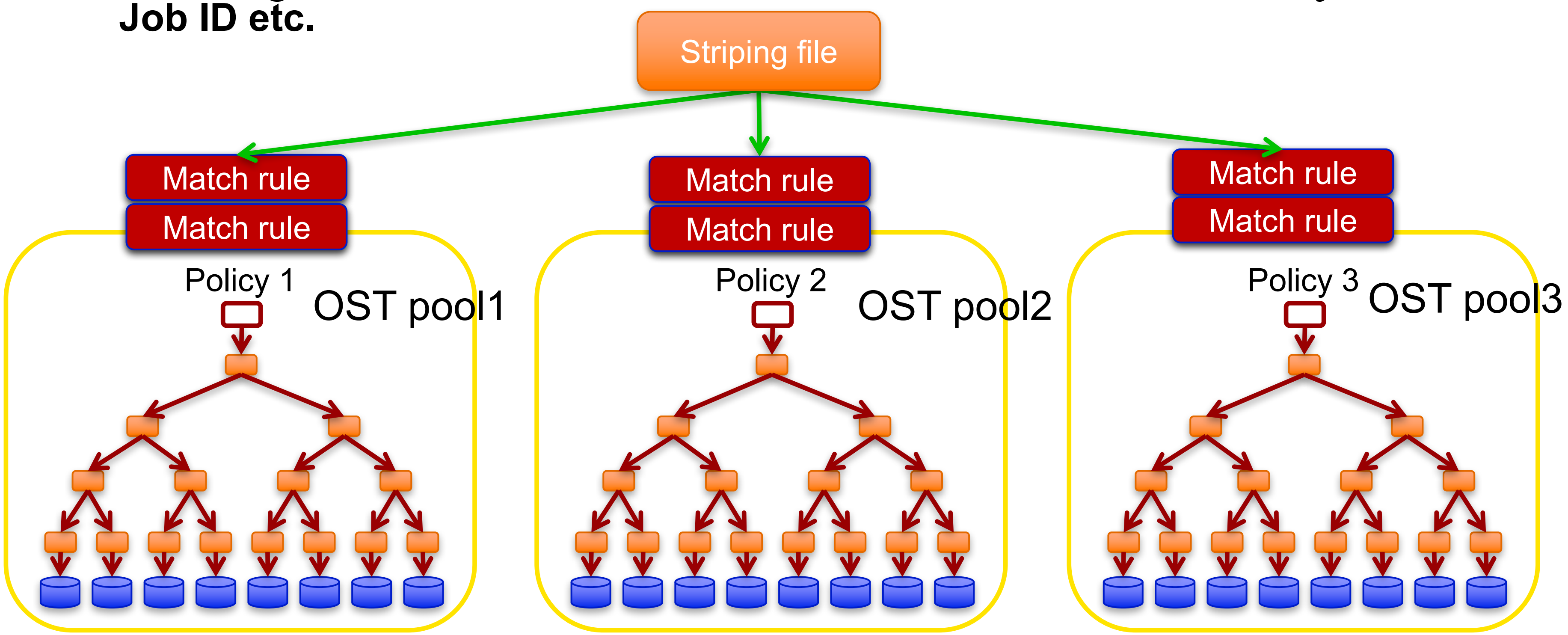
Implementation: RTDS Tree

W[i]: Weight of OST i
S[i]: W[0] + W[1] + … + W[i]



OST Pool + RTDS

- One RTDS tree is generated for each OST pool
- Pool is currently inherited from parent, we want to choose pool according to a policy
- Each OST pool has a series of match rules
- File will locate on an OST pool if the rules of the pool are matched
- Matching rules are based on file attributes like UID, GID, NID, Project ID, Job ID etc.



Relative Weights Between OSTs

- **What is relative weight?**
  - When a OST is being selected as one of the stripes of a file, the weight of another (or the same) OST will be updated accordingly in the next round
- **W[i]: Weight of OST i**
- **RW(i, j)**
  - Describes how OST i affects OST j
  - When OST i is being selected, then before next round, W[j] will be changed to W[j] * RW(i, j)

Implementation of relative weight

- **Allocation of file with multiple stripes**
  - 1. Copy the weight array from the public weight array. All allocation processes shares the same RTDS tree though.
  - 2. Allocate an object according to the current weight array
  - 3. Update the private weight array according to the relative weights
  - 4. Go to step 2 to allocate the next object
- **Allocation of file with only one stripe**
  - Use shared public weight array, no need to copy one

Configuration Examples of Relative weight

- Set RW(i, i) to 0, to avoid allocating more than one objects on OST i for a single file
- Set RW(i, j) to 1 when i != j, if the OSTs are considered unrelated
- Set RW(i, j) < 1, if OST i and OST j are on the same OSS, and we want to try to avoid locating two stripes on the same OSS
- Set RW(i, j) > 1, if we want to locate the next object on OST j which have the same specification(e.g. SSD/HDD based OSTs) with OST i
- Usual values of RW: 0, 1/2, 1, 2, INFI, etc

Daemon of weight adjusting

- **A daemon should be monitoring the system and adjusting the weights of all OSTs from time to time**
  - The weights will be adjusted according to free spaces, free bandwidth, inodes, etc.
  - The weights will be updated every one minute or so
  - Smart algorithms or AI can be used for the dynamical adjusting process
- **Example of dynamical configuration adjustment**
  - LIME: Lustre Intelligent Management Engine
  - https://github.com/DDNStorage/Lime
  - Collects the real-time performance statistics of a job
  - Changes the TBF rates every one second to provide QoS guarantees or enforce performance limitations

Use cases

- **Quick space balance when adding new OSTs**
  - Configure empty OSTs with higher weights than full OSTs
- **Load balance between OSTs**
  - Configure idle OSTs with higher weights than busy OSTs
- **Avoid to use degraded OSTs**
  - Configure the OSTs that are doing RAID rebuilding with zero weight
- **Reserved quick OSTs for high-priority jobs**
  - Separate OSTs into OST pools according to speed
  - Define matching rules to separate jobs by priority levels
- **Advanced QoS management together with NRS TBF policy**
  - The bandwidth of OSTs can be allocated by using TBF and RTDS together

Advices?

- https://jira.hpdd.intel.com/browse/LU-9809
- https://review.whamcloud.com/28292

Thank you!