**SSM Erasure Coding (EC) Support on HDFS 3.x**

* **Background & Motivation**

SSM provides a comprehensive hot/cold data optimized solution. It can collect a variety of properties for HDFS data, such as file path, file size, access time, modification, storage policy, access count, etc. In SSM rule, users can define their desired operations and conditions for triggering operations or so called actions in SSM by using data properties.

Erasure Coding (EC) is designed for cold data storage in HDFS. The storage space can be saved greatly if the cold data is stored in EC way compared with the replication way. HDFS has not realized the conversion for a file from replication to EC directly. It can only make newly created file be stored in a desired way by setting EC policy for parent directory beforehand. SSM is designed to provide the conversion between any pair of different EC polices. For a SSM rule with EC action embedded, SSM can track user-defined condition for cold data and make the conversion occur once the condition is met. For an instance, user can submit a SSM rule to monitor files under a directory. For each file, if it is accessed less than 3 times in 30 days, SSM will convert it to the one stored in EC way with a given or default EC policy.

* **SSM EC Related Actions**

Action is a kind of operation executed by SSM. It can be submitted by user and then executed immediately. Also, action can be embedded into SSM rule and will be executed only if the rule condition is met. We designed the following EC related actions: EC action, unEC action, reset EC action and check EC action.

1. **SSM EC action: ec -file $path [-policy $policy]**

* **[dir case]** If $path is a dir, set its EC policy, which will have no impact on existed file but can make new created file stored in EC way henceforward.
* **[file case]** If $path is a file, SSM will open the file with append permission for locking file and then read permission for read. The file content will be written into a temp file in EC way. The temp file is named by following HDFS’s convention for temp file. The file attributes (owner, permission, etc.) will be kept consistent. After the conversion is finished, the EC file will be renamed to original one. So the original file will be overwritten.
* **[EC policy]** If -policy is not given, a default policy will be used. User can configure default policy in HDFS conf directory which is already available for SSM use in our realization. HDFS has provided REPLICATION as a special policy. So if REPLICATION is given as EC policy in EC action, SSM will convert the file to the one stored in replication way. For another case, if the submitted EC action is given a same policy as this file’s current one, SSM will not and has no need to execute the conversion operation.

1. **SSM unEC action: unec -file $path**

* **[dir case]** If $path is a dir, just set the policy to REPLICATION.
* **[file case]** If $path is a file, SSM will go to the execution path of an EC action with REPLICATION policy.

1. **SSM reset EC action: resetec -file $path**

* **[dir case]** If $path is a dir, just set the policy to the closest EC policy in its parent dir chain.
* **[file case]** If $path is a file, SSM will check whether the current policy is as same as the closest EC policy in its parent dir chain. Only if they are not same, SSM will execute an EC action to make the conversion and also keep file attributes consistent.

1. **SSM check EC action: checkec -file $path**

* Show EC policy for $path.
* **SSM EC Actions Failures**
* If an EC, unEC or reset EC action fails in intermediate stage (during EC), the temp file should be deleted and keep everything in HDFS as same as that before action execution.
* **EC Workload Constraint [An optimization]**
* EC can consume large resources, especially network IO. We will consider to add a throttle to avoid cluster overload. When a threshold is reached, SSM should delay the dispatching of more EC action if any.