

Support Dynamic Logger Level Adjustment

Jira: <https://issues.apache.org/jira/browse/FLINK-33320>
[FLIP-388](#)

Motivation

During the process of routine program debugging or troubleshooting, analyzing system logs is a common approach.

Comprehensive and detailed system logs contribute to improved visibility of internal system execution information and also enhance the efficiency of program debugging or issue troubleshooting. However, comprehensive and detailed log settings can lead to the following issues:

1. A sharp increase in log volume, accelerating disk occupancy.
2. Potential risks of system performance degradation due to a large volume of log printing.
3. The need to simplify log configuration subsequently.

Therefore, introducing a mechanism to dynamically adjust the online log output level in the event of diagnosing online issues or debugging programs could be meaningful.

This mechanism should ideally provide the following two basic capabilities:

1. Dynamically adjust log levels.
2. Query the current log levels of the JM/TM in the cluster.

Pre-research for log framework

These examples are primarily used to document the process of dynamically adjusting logs for each logging framework and to illustrate the feasibility of dynamically adjusting log levels.

- slf4j & log4j1:
<https://github.com/RocMarshal/dynamic-logger-demo/tree/dev/slf4j-demo/slf4j-log4j1-demo>
- slf4j & log4j2:
<https://github.com/RocMarshal/dynamic-logger-demo/tree/dev/slf4j-demo/slf4j-log4j2-demo>
- slf4j & logback:
<https://github.com/RocMarshal/dynamic-logger-demo/tree/dev/slf4j-demo/slf4j-logback-demo>
- Mixed compile for **slf4j** & **[log4j1/log4j2/logback]**:
<https://github.com/RocMarshal/dynamic-logger-demo/tree/dev/slf4j-demo/slf4j-three-impls-demo>

Proposed Change

- Why only for slf4j & **slf4j** & **[log4j1/log4j2/logback]** ?
Because the Flink engine uses the bridge interface of Slf4j internally
- Re registration of TM
If the current RM has already performed a dynamic log adjustment operation, then the newly registered TM will also perform a log change operation
- Changes and query interfaces in HA mode do not take effect on the JM component of the slave role

Public change

- /put-loggerLevel
 - METHOD: PUT
 - **Response code:** 200 OK
 - Request:

```
{
    loggerLevel: {
        "root": "DEBUG",
        "akka.xxx": "INFO", ...
    }
}
```
 - Response:

```
{}
```
- /get-loggerLevel
 - METHOD: GET
 - **Response code:** 200 OK
 - Request:

```
{}
```
 - Response:

```
{
    "JobManager": {
        "jm-1@xx.xx.xx.xx": {
            "rootLogger": "INFO",
            .....
        }
    },
    "TaskManager": {
        "tm-1@xx.xx.xx.xx": {
            "rootLogger": "INFO",
            .....
        },
        .....
    }
}
```

Non-public change

- Change for 'put-loggerLevel'
 - Add the rpc method for ResourceManagerGateway

```
CompletableFuture<List<Acknowledge>> changeLogLevel(  
    @Nonnull ChangeLogLevelRequest changeLogLevelRequest);
```
 - Add the rpc method for TaskExecutorGateway

```
CompletableFuture<Acknowledge> changeLogLevel(  
    @Nonnull ChangeLogLevelRequest request);
```
 - Introduce a class named ChangeLogLevelRequest

```
class ChangeLogLevelRequest implements Serializable{  
    Map<String, String> loggerLevel;  
    // other placeholders...  
}
```
- Change for 'get-loggerLevel'
 - Add the rpc method for ResourceManagerGateway

```
CompletableFuture<Map<Serializable, Serializable>>  
getLogLevel();
```
 - Add the rpc method for TaskExecutorGateway

```
CompletableFuture<Map<String, String>> getLogLevel();
```

References

- <https://logging.apache.org/log4j/1.2/>
- <https://logback.qos.ch/>
- <https://logging.apache.org/log4j/2.x/>
- <https://www.slf4j.org/>
- <https://commons.apache.org/proper/commons-logging/>

Acknowledgements

Thanks for the help from fanrui.