[Livy Client](http://wiki.intra.xiaojukeji.com/display/BigdataOfflineEngine/Livy+Client+Detail+Designing+Document)

**Brief Introduction**

Livy-Client is a kind of higher-layer spark client, User execute shell script with Livy-Client like using script provided by spark client. Livy-Client parse all parameters in script and filled into rest template and submit to livy server. The difference between Livy-Client and spark client is Livy-Client submit spark job to livy server and user don't need to install a spark client in local machine. Livy-Client is designed to replace existing spark client which may have different version and spread on each user machine, user can submit spark job by script as before, and these spark jobs are monitored and managed by livy. In order to minimize user's perception about change spark client to Livy-Client, Livy-Client adapt most of all parameters in spark client scripts and livy rest api, and because of using livy to submit jobs, user can fully enjoy Livy's security verification, session management, multi-tenancy and other functions.

**Advantages**

* Compared with spark client, Livy-Client will almost NOT update which spent patient of most user, backend livy server can change spark dependency at realtime and user has no need to know that.
* Livy-Client is more lightweight than spark client and has no dependency, moreover spark job running in cluster mode will not occupy memory and calculation resources in local machine.
* All jobs submited by Livy-Client will run in cluster mode which is more convenient to shoot the trouble. Livy-Client will get job progress and print like spark client.

**Architecture**

Livy-Client adapt input in submit script like spark-client, and the underlying access to livy by rest api, to submit job and get job progress or result. backend livy server control all spark level execution.

when user use different Livy-Client, actually start different kind of Livy Interpreter(interactive), livy submit matched spark-submit and start a SparkYarnApp in livy.

There are 4 Livy-Clients correspond to 4 spark-submit script:

* livy-sql corresponding to spark-sql, using Livy SQLInterpreter to execute code.
* livy-shell corresponding to spark-shell, using Livy SparkInterpreter to execute code.
* livy-pyspark corresponding to pyspark, using Livy PythonInterpreter to execute code.
* livy-submit corresponding to spark-submit, using Livy SparkYarnApp to monitor and manage job.

these 4 clients is used as shown in Figure 1:

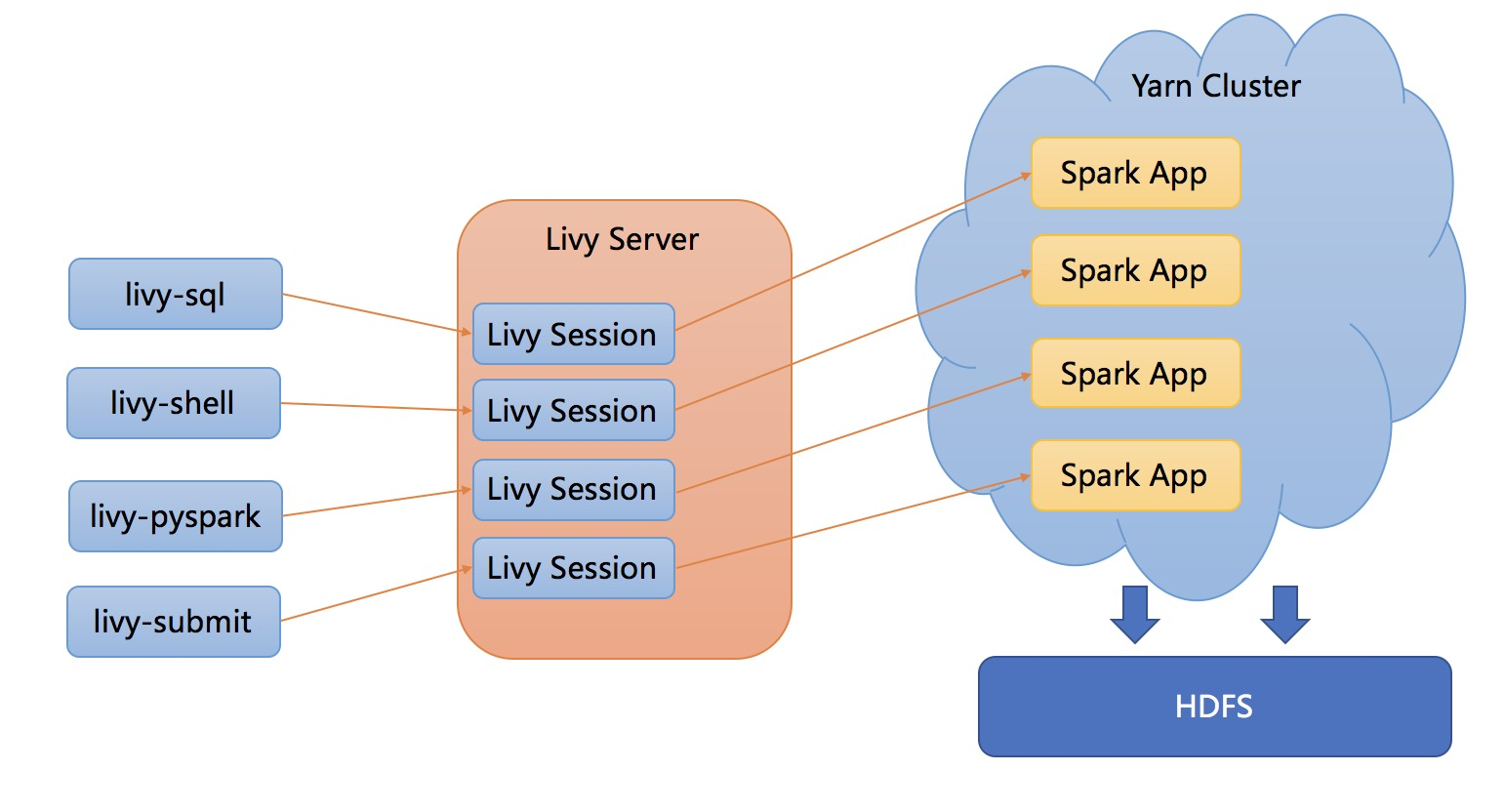


Figure 1

* Consistency：All Spark jobs are execute in cluster mode, and all used livy server's spark dependency, The execution result is consistent.
* Isolation：Jobs submitted in different client will match different sessions on livy server, and every session use its own sparkContext to make sure different jobs are isolated.
* Synchronization：Livy-Client use polling method in creating livy session and obtaining execution result until creation or execution is completed.

There are these steps in Livy-Client submitting a spark job:  **Submit CommandLine, Parsing parameters, Separate varient params, Start Interpreter, Load hiveconf, Execute code, polling for result, print result.**as Figure 2：

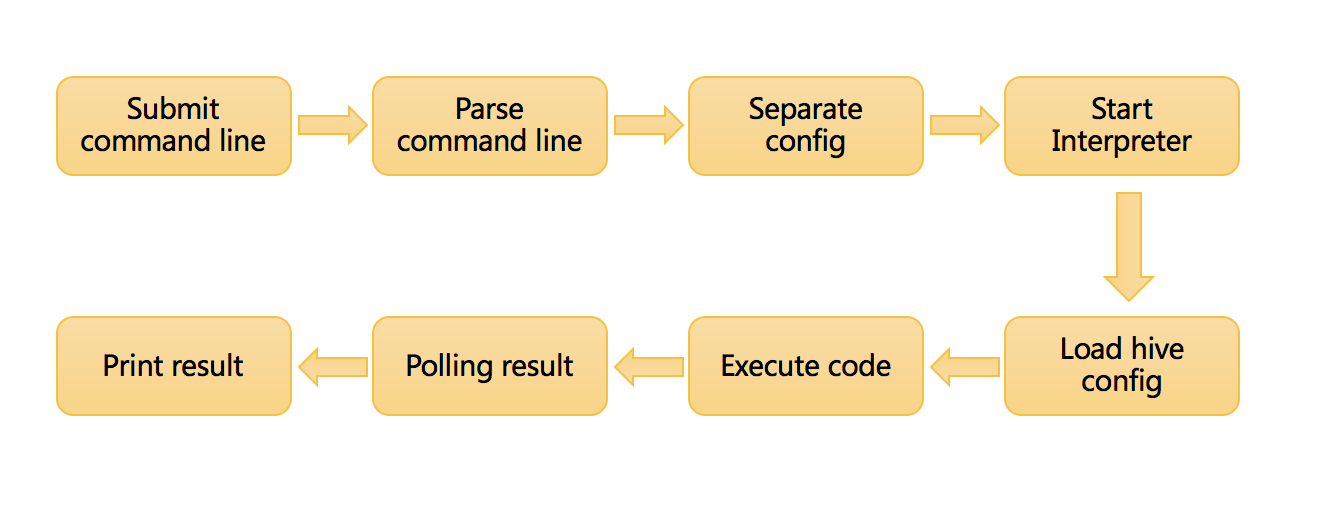


Figure 2

**Submit CommandLine**：in commandLine, use abbreviation to set common spark or livy params, use --conf to set customized spark params or livy rsc params, and use --hiveconf to set hive params. you can input --help for user guidance.

**Parsing parameters**：use parser like spark-submit to parse command line and init SparkSubmitArgument class.

**Separate varient params**：separate spark config, livy config and hive config in SparkSubmitArgument.

**Start interpreter**：start LivyInterpreter in Livy-Client, at the same time submit a rest request to create session in livy server and LivyInterpreter control this session by sessionId which returned from rest api respond.

**Load hiveconf**：if script contains hiveconf config，use SET command to active config after LivyInterpreter has been started.

**Execution code**：user can use -e or -f param to make client parse code line by line, or input code interactively. all code will be contained in rest request, livy server receive request and start statement. when execution, Livy Client will get sparkUiUrl from livy session info, Client will get job or stage progress from sparkui, and print in console.

**polling result**：Livy-Client use sessionId and statementId to construct rest request access to livy, get execution result until complete.

**print result**：when statement is in finish state, Livy-Client will print statement result field in console, livy-submit will not print result, only progress.

**Module design**

The workflow of Livy Client is marked as Figure 3：

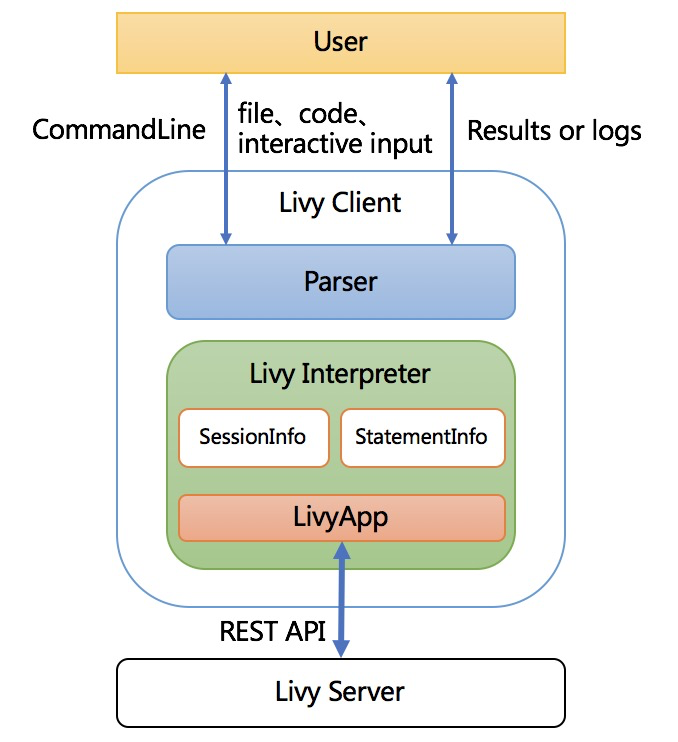


Figure 3

Client

we supply 4 kind of Livy-Clients which is like spark client script——livy-submit, livy-sql, livy-shell and livy-pyspark. and usage is the same as spark script.

we need set several livy params in client to construct rest template, as list：

| **param name** | **introduction** |
| --- | --- |
| --proxy-user | set proxyuser |
| --queue | set queue |
| --url | connect to this livy server |
| --database | link to which database |
| --conf | set custom spark or livy.rsc configs |

Parser and Adaptor

Like spark-submit Livy-Client use parser to reflect args in commandLine into attributes of SparkSubmitArgument class, and use a Map to package all parsed args and transfer to Livy Client class.

* separate spark configs and livy configs
* identify hive configs
* support set all available spark configs by –conf

Livy Client Class

* The main function of Livy Client Class is uploading local files to HDFS(hadoop dependency), set params, read code from file or interactive input。
* when start LivyInterpreter，load local initFile or set hive configs.
* like hive commandLine，Livy-Client not only support -e to submit a sql string or -f to submit a code file,  but also support interactive input.
* support user upload local files to HDFS, and release them when session closed.

Livy Interpreter

Each time using Livy-Client to submit a spark job(except livy-submit), A LivyInterpreter will be created and inited，when Livy Interpreter is initializing, LivyInterpreter construct request params into rest template, and send to livy server to create a new session. Livy-Client polling until session has been created, and get session info into Livy-Client SessionInfo class.

When execute code, LivyInterpreter will get state field in session info and check whether this is a available session. if it does, LivyInterpreter construct new rest template, send user code to livy server and start statement to execute.

Livy-Client will polling to get statement state, when state is in finish state and state is success, Livy-Client unserialize the json string into statementInfo，and print result field.

Livy App

* Use http client to init RestTemplate, and implement livy rest api
* Implement method to upload local files which is called by Livy Client Class

Livy Batch

Livy batch parse args and package them into rest template, send to livy server, create spark-submit process by livy.

Progress module

job submitted by spark-submit in cluster mode cannot get job and stage info, and livy ui also has no progress info in batch.

The spark ui url and driver log url of spark application are preserved in Livy-Client SessionInfo class, The progress module get spark app stage info from spark ui url and print progress in console.

Livy-Client in Didi

Livy-Client is used to replace existing spark client in some data dev platform(10+ clients) and personal users, Spark job submitted by Livy-Client is 600+ per day and increase rapidly.