Master Server Application (main)

***Libraries required in the classpath:***

* org.osgi.core.jar
* org.restlet.jar
* javax.servlet.jar
* org.eclipse.jetty.ajp.jar
* org.eclipse.jetty.continuations.jar
* org.eclipse.jetty.http.jar
* org.eclipse.jetty.io.jar
* org.eclipse.jetty.server.jar
* org.eclipse.jetty.util.jar
* org.jsslutils.jar
* org.restlet.ext.jetty.jar
* org.restlet.ext.ssl.jar
* commons-logging-1.1.1.jar
* log4j-1.2.9.jar
* microsoft-windowsazure-api-0.2.0.jar *(for windows azure connection)*
* commons-lang3-3.1.jar *(for windows azure connection)*
* jackson-jaxrs-1.8.3.jar *(for windows azure connection)*
* jackson-mapper-asl-1.8.3.jar *(for windows azure connection)*
* jackson-xc-1.8.3.jar *(for windows azure connection)*
* javax.inject-1.jar *(for windows azure connection)*
* jaxb-impl-2.2.3-1.jar *(for windows azure connection)*
* jersey-client-1.10-b02.jar *(for windows azure connection)*
* jersey-core-1.10-b02.jar *(for windows azure connection)*
* jersey-json-1.10-b02.jar *(for windows azure connection)*
* jettison-1.1.jar *(for windows azure connection)*
* jackson-core-asl-1.8.3.jar *(for Windows Azure & Amazon S3 connection)*
* stax-api-1.0.1.jar *(for Windows Azure & Amazon S3 connection)*
* aws-java-sdk-1.2.2.jar *(for Amazon S3 connection)*
* commons-codec-1.4.jar *(for Amazon S3 connection)*
* httpclient-4.1.1.jar *(for Amazon S3 connection)*
* httpcore-4.1.jar *(for Amazon S3 connection)*
* mail-1.4.3.jar *(for Amazon S3 connection)*
* stax-1.2.0.jar *(for Amazon S3 connection)*

***Files required in the classpath:***

* commons-logging.properties
* log4j.properties
* restlet-logging.properties
* MasterConfig.properties
* IDAProp.properties

***Class:*** sg.edu.nyp.sit.svds.master.Main

***Parameters:***

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| type | Type of the master server to run. Accepts either “main” or “secondary”. | *–type main* |
| ida\_config | Relative/Absolute path to the IDA configuration properties file | *–ida\_config IDAProp.properties* |
| sys\_config | Relative/Absolute path to the master properties file | *–sys\_config MasterConfig.properties* |

***Format:***

sg.edu.nyp.sit.svds.master.Main –type main –<parameter name 1> <parameter value 1> –<parameter name 2> <parameter value 2> … –<parameter name n> <parameter value n>

***Note:***

* The port on which the master application will be running and the path to the image and logs are configured in the master properties file.
* The path to the image and logs must exist and given read and write *(including able to delete)* permissions.
* The value in the “client.master.\*” property of the *svdsclient.properties* file *(under sg.edu.nyp.sit.svds.client)* must be change according to the IP/host name and the port number *(if applicable)* where the master server application is deployed.
* To use the *restlet-logging.properties* file, set the JVM property “-Djava.util.logging.config.file” during runtime with the “java” command.

*For sample windows batch file, refer to runMaster(sample)\_v2.bat.*

Master Server Application (secondary)

***Libraries required in the classpath:***

* commons-logging-1.1.1.jar
* log4j-1.2.9.jar

***Files required in the classpath:***

* commons-logging.properties
* log4j.properties
* restlet-logging.properties
* SecondaryMasterConfig.properties

***Class:*** sg.edu.nyp.sit.svds.master.Main

***Parameters:***

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| type | Type of the master server to run. Accepts either “main” or “secondary”. | *–type secondary* |
| sys\_config | Relative/Absolute path to the secondary master properties file | *–sys\_config SecMasterConfig.properties* |

***Format:***

sg.edu.nyp.sit.svds.master.Main –type secondary –<parameter name 1> <parameter value 1> –<parameter name 2> <parameter value 2> … –<parameter name n> <parameter value n>

***Note:***

* Read and write *(including able to delete)* permission on the temporary directory to be given to the user running the application.
* Properties like the location of the main master application to be configured in the secondary master properties file.
* To use the *restlet-logging.properties* file, set the JVM property “-Djava.util.logging.config.file” during runtime with the “java” command.

File Slice Store Application

***Libraries required in the classpath:***

* org.osgi.core.jar
* org.restlet.jar
* javax.servlet.jar
* org.eclipse.jetty.ajp.jar
* org.eclipse.jetty.continuations.jar
* org.eclipse.jetty.http.jar
* org.eclipse.jetty.io.jar
* org.eclipse.jetty.server.jar
* org.eclipse.jetty.util.jar
* org.jsslutils.jar
* org.restlet.ext.jetty.jar
* org.restlet.ext.ssl.jar
* commons-logging-1.1.1.jar
* log4j-1.2.9.jar

***Files required in the classpath:***

* commons-logging.properties
* log4j.properties
* restlet-logging.properties
* SliceStoreConfig.properties

***Class:*** sg.edu.nyp.sit.svds.filestore.Main

***Parameters:***

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| id | Unique name given to the slice store application. Accepts only alpha-numeric and underscores. | *–id FS1* |
| fport | Port number that application would listen on for file slice requests | *–fport 8010* |
| sport | Port number that application would listen on for slice store status requests | *–sport 8011* |
| host | Host name or the IP address of the host running the application. Used to bind with the server, can be internal IP address. If none is supplied then restlet takes the default given by machine | *–host localhost* |
| reghost | Host name or the IP address of the host running the application. Used to register with the server, can be public IP address. Only applicable if the value for reg parameter is “0”. | *-reghost 175.12.134.15* |
| path | Absolute location for creating and reading the file slices. | *–path D:\\ DiffusedCloudStorage\\FS1* |
| config | Relative/Absolute path to the slice store properties file | *–config SliceStoreConfig.properties* |
| reg | Value to determine if registration with master server is required:   * 0 – register with master server and periodically ping back (default value if not specified) * 1 – Do not register with master server but still periodically ping back * 2 – Do not register with master server and do not ping back | *-reg 2* |

***Format:***

sg.edu.nyp.sit.svds.filestore.Main –<parameter name 1> <parameter value 1> –<parameter name 2> <parameter value 2> … –<parameter name n> <parameter value n>

***Note:***

* The path provided to the application must exist.
* Read and write *(including able to delete)* permission on the path provided to the application to be given to the user running the application.
* Location of master application and namespace supported must be configured in *sliceStoreConfig.properties.*
* If the status operations requires SSL/TLS, then the “CN” property in the server certificate specified in the properties file must match the value of the host parameter.
* To use the *restlet-logging.properties* file, set the JVM property “-Djava.util.logging.config.file” during runtime with the “java” command.

*For sample windows batch file, refer to runFS(sample)\_v2.*

Proxy Server Application

***Libraries required in the classpath:***

* org.osgi.core.jar
* org.restlet.jar
* javax.servlet.jar
* org.eclipse.jetty.ajp.jar
* org.eclipse.jetty.continuations.jar
* org.eclipse.jetty.http.jar
* org.eclipse.jetty.io.jar
* org.eclipse.jetty.server.jar
* org.eclipse.jetty.util.jar
* org.jsslutils.jar
* org.restlet.ext.jetty.jar
* org.restlet.ext.ssl.jar
* commons-logging-1.1.1.jar
* log4j-1.2.9.jar
* sqlite-jdbc-3.7.2.jar
* mysql-connector-java-5.1.20-bin.jar

***Files required in the classpath:***

* commons-logging.properties
* log4j.properties
* restlet-logging.properties
* ProxyConfig.properties

***Class:*** sg.edu.nyp.sit.pvfs.proxy.Main

***Parameters:***

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| port | Port number that application would listen on for request | *–port 6010* |
| path | Absolute location for saving the SQLite database sent by Android application | *–path D:\\ DiffusedCloudStorage\\FS1* |
| sys\_config | Relative/Absolute path to the proxy properties file | *–sys\_config ProxyConfig.properties* |

***Format:***

sg.edu.nyp.sit. pvfs.proxy.Main –<parameter name 1> <parameter value 1> –<parameter name 2> <parameter value 2> … –<parameter name n> <parameter value n>

***Note:***

* The path provided to the application must exist.
* Read and write *(including able to delete)* permission on the path provided to the application to be given to the user running the application.
* Subscriber database (MySQL) must be configured in *ProxyConfig.properties.*
* If the status operations requires SSL/TLS, then the “CN” property in the server certificate specified in the properties file must match the value of the ssl.address parameter.
* To use the *restlet-logging.properties* file, set the JVM property “-Djava.util.logging.config.file” during runtime with the “java” command.

*For sample windows batch file, refer to runProxy(sample)\_v2.*

File Load Test

***Libraries required in the classpath:***

* org.osgi.core.jar ***\****
* org.restlet.jar ***\****
* javax.servlet.jar ***\****
* org.eclipse.jetty.ajp.jar ***\****
* org.eclipse.jetty.continuations.jar ***\****
* org.eclipse.jetty.http.jar ***\****
* org.eclipse.jetty.io.jar ***\****
* org.eclipse.jetty.server.jar ***\****
* org.eclipse.jetty.util.jar ***\****
* org.jsslutils.jar ***\****
* org.restlet.ext.jetty.jar ***\****
* org.restlet.ext.ssl.jar ***\****
* commons-logging-1.1.1.jar
* log4j-1.2.9.jar
* microsoft-windowsazure-api-0.2.0.jar *(for windows azure connection)*
* commons-lang3-3.1.jar *(for windows azure connection)*
* jackson-jaxrs-1.8.3.jar *(for windows azure connection)*
* jackson-mapper-asl-1.8.3.jar *(for windows azure connection)*
* jackson-xc-1.8.3.jar *(for windows azure connection)*
* javax.inject-1.jar *(for windows azure connection)*
* jaxb-impl-2.2.3-1.jar *(for windows azure connection)*
* jersey-client-1.10-b02.jar *(for windows azure connection)*
* jersey-core-1.10-b02.jar *(for windows azure connection)*
* jersey-json-1.10-b02.jar *(for windows azure connection)*
* jettison-1.1.jar *(for windows azure connection)*
* jackson-core-asl-1.8.3.jar *(for Windows Azure & Amazon S3 connection)*
* stax-api-1.0.1.jar *(for Windows Azure & Amazon S3 connection)*
* aws-java-sdk-1.2.2.jar *(for Amazon S3 connection)*
* commons-codec-1.4.jar *(for Amazon S3 connection)*
* httpclient-4.1.1.jar *(for Amazon S3 connection)*
* httpcore-4.1.jar *(for Amazon S3 connection)*
* mail-1.4.3.jar *(for Amazon S3 connection)*
* stax-1.2.0.jar *(for Amazon S3 connection)*

***Files required in the classpath:***

* commons-logging.properties
* log4j.properties
* svdsclient.properties
* restlet-logging.properties
* MasterConfig.properties ***\****
* IDAProp.properties ***\****
* SliceStoreConfig.properties ***\****

***Class:*** sg.edu.nyp.sit.svds.client.FileLoadTestApp

***Parameters:***

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| clients | No of clients thread to run. | *–clients 2* |
| filesize | Size of the file in bytes that each client thread will attempt to write and read. | *–filesize 1048576* |
| mode | If the application is using the SVDS file (value is “SVDS”) or java IO file (value is “NFS”) functionalities to perform the load test. | *–mode SVDS* |
| nfspath | Directory path (not ending with separator) where the load test files will be created.  Applicable only if the mode is “NFS”. | *–nfspath .* |
| streaming | If the client threads should write and read in streaming mode. Accepts either “yes” or “no”.  Applicable only if the mode is “SVDS”. | *–streaming no* |
| localmasterconfig | Relative/Absolute path to the directory containing the master properties files. The files must be named “MasterConfig.properties” and “IDAProp.properties”.  Applicable only if the mode is “SVDS” and the master server application and the slice store servers are to be started by the load test application as well.  Mutually exclusive with “master” property. | *–localmasterconfig .* |
| master | Address and port number *(if applicable)* of the master server application that accept slice store registration.  Applicable only if the mode is “SVDS” and the slice store servers are to be started by the load test application but the master server application is already started by other means.  Mutually exclusive with the “localmasterconfig” properties. | *–master localhost:9010* |
| slicestores | No of slice store servers to start up.  If the slice stores server are already started by other means, omit this property.  Note that all slice stores have to be register to the master server application used in the load test application.  Applicable only if the mode is “SVDS”. | *–slicestores 3* |
| slicestorespath | Absolute location for creating and reading the file slices by each slice store server application. For each slice store application started by the load test application, a folder will be created in the path provided.  Must exist together with “slicestores” property.  Applicable only if the mode is “SVDS” | *–slicestorespath D:\\ DiffusedCloudStorage\\storage* |
| slicestoreconfig | Relative/Absolute path to the slice store properties file.  Must exist together with “slicestores” property.  Applicable only if the mode is “SVDS”. | *–slicestoreconfig*  *.\\SliceStoreConfig.properties* |

***Format:***

Running only the client (SVDS mode)

sg.edu.nyp.sit.svds.filestore.Main –clients <parameter value> –filesize <parameter value> –mode SVDS –streaming <parameter value>

Running the client with both the slice stores & master server (SVDS mode)

sg.edu.nyp.sit.svds.filestore.Main –mode SVDS –clients <parameter value> –filesize <parameter value>

–streaming <parameter value> –localmasterconfig <parameter value> –slicestores <parameter value> –slicestorespath <parametervalue> –slicestoreconfig <parameter value>

Running the client with the slice stores (SVDS mode)

sg.edu.nyp.sit.svds.filestore.Main –mode SVDS –clients <parameter value> –filesize <parameter value>

–streaming <parameter value> –master <parameter value> –slicestores <parameter value> –slicestorespath <parametervalue> –slicestoreconfig <parameter value>

Running in NFS mode

sg.edu.nyp.sit.svds.filestore.Main –clients <parameter value> –filesize <parameter value> –mode NFS –nfspath <parameter value>

***Note:***

* All path(s) provided to the application must exist.
* Read and write *(including able to delete)* permission on the path provided to the application to be given to the user running the application.
* Slice store servers must be started by the load test application is the master server application is to be started by the load test application as well because the slice stores server must registered themselves to the master server application.
* The value in the “client.master.\*” property of the *svdsclient.properties* file *(under sg.edu.nyp.sit.svds.client)* is to be change according to the IP/host name and the port number *(if applicable)* where the master server application is deployed.
* To use the *restlet-logging.properties* file *(when master and/or slice store application is run together with the client)*, set the JVM property “-Djava.util.logging.config.file” during runtime with the “java” command.

***\**** *Optional if running only the client*

*For sample windows batch file, refer to runLoadTest\_onlyClient(sample)\_v2.bat, runLoadTest\_full(sample)\_v2.bat and/or runLoadTest\_onlyFS(sample)\_v2.bat*

*or*

*refer to runLoadTest\_nfs(sample).bat.*

Windows Azure SliceStore Registration

***Libraries required in the classpath:***

* commons-logging-1.1.1.jar
* log4j-1.2.9.jar
* microsoft-windowsazure-api-0.2.0.jar
* commons-lang3-3.1.jar
* jackson-core-asl-1.8.3.jar
* jackson-jaxrs-1.8.3.jar
* jackson-mapper-asl-1.8.3.jar
* jackson-xc-1.8.3.jar
* javax.inject-1.jar
* jaxb-impl-2.2.3-1.jar
* jersey-client-1.10-b02.jar
* jersey-core-1.10-b02.jar
* jersey-json-1.10-b02.jar
* jettison-1.1.jar
* stax-api-1.0.1.jar

***Files required in the classpath:***

* commons-logging.properties
* log4j.properties

***Class:*** sg.edu.nyp.sit.svds.master.filestore.AzureSliceStoreRegistration

***Parameters:***

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| [0] | Absolute path to the file containing azure storage information. Generally follows the properties file format, for more details, refer below. | *D: \\DiffusedCloudStorage\\resource \\azureSliceStores(sample).txt* |
| [1] | URL of the master server. | *localhost:9010* |
| [2] | Connector to the master server. Use “http” for normal connections, or “https” for SSL connections. | *http* |
| [3] | Optional. Absolute or relative path to the key store containing the certificate to the master server if  connection to the master server is using “https”. | *moeifks.jks* |
| [4] | Password to the key store specified in [3]. If [3] is not specified, this is to be omitted as well. |  |
| [5] | Type of the key store specified in [3]. If [3] is not specified, this is to be omitted as well. | *JKS* |

***Format:***

sg.edu.nyp.sit.svds.filestore.AzureSliceStoreRegistration <parameter [0] value> <parameter [1] value> <parameter [2] value> <parameter [3] value> <parameter [4] value> <parameter [5] value>

***Format of file in parameter [2]:***

azure.cnt=*<no of entries in the file>*

azure.id**0**=*<id of the slice store, must be at between 3-63 characters>*

azure.url**0**=*<url of the azure storage, usually http\://blob.core.windows.net/>*

azure.namespace**0**=*<namespace to register the slice store to, separate multiple namespace with comma>*

azure.storageaccount**0**=*<account name of the azure storage>*

azure.secretaccesskey**0**=*<secret access key of the azure storage>*

azure.usedevelopment**0**=*<if the slice store implementation should use the development azure storage URL or the normal storage URL. 0=false, 1=true>*

. . .

azure.id**<n>**=…

azure.url**<n>**=…

azure.namespace**<n>**=…

azure.storageaccount**<n>**=…

azure.secretaccesskey**<n>**=…

azure.usedevelopment**<n>**=…

***Note:***

* **<n>** refers to the number of the slice store registrations in the file, start from 0.
* azure.idof each slice store is the container name in windows azure storage.
* To use the *restlet-logging.properties* file, set the JVM property “-Djava.util.logging.config.file” during runtime with the “java” command.

*For sample registration file, refer to azureSliceStores(sample).txt.*

*For sample windows batch file, refer to registerAzureFS.bat.*

Amazon S3 SliceStore Registration

***Libraries required in the classpath:***

* commons-logging-1.1.1.jar
* log4j-1.2.9.jar
* httpclient-4.1.1.jar
* httpcore-4.1.jar
* aws-java-sdk-1.2.2.jar
* commons-codec-1.4.jar
* jackson-core-asl-1.8.3.jar
* mail-1.4.3.jar
* stax-1.2.0.jar
* stax-api-1.0.1.jar

***Files required in the classpath:***

* commons-logging.properties
* log4j.properties

***Class:*** sg.edu.nyp.sit.svds.master.filestore.S3SliceStoreRegistration

***Parameters:***

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| [0] | Absolute path to the file containing S3 storage information. Generally follows the properties file format, for more details, refer below. | *D: \\DiffusedCloudStorage\\resource \\* *s3SliceStores(sample).txt* |
| [1] | URL of the master server | *localhost:9010* |
| [2] | Connector to the master server. Use “http” for normal connections, or “https” for SSL connections. | *http* |
| [3] | Optional. Absolute or relative path to the key store containing the certificate to the master server if  connection to the master server is using “https”. | *moeifks.jks* |
| [4] | Password to the key store specified in [3]. If [3] is not specified, this is to be omitted as well. |  |
| [5] | Type of the key store specified in [3]. If [3] is not specified, this is to be omitted as well. | *JKS* |

***Format:***

sg.edu.nyp.sit.svds.master.filestore.S3SliceStoreRegistration <parameter [0] value> <parameter [1] value> <parameter [2] value> <parameter [3] value> <parameter [4] value> <parameter [5] value>

***Format of file in parameter [2]:***

s3.cnt=*<no of entries in the file>*

s3.id**0**=*<id of the slice store, must be at between 3-63 characters>*

s3. bucket0=*<name of the bucket in the S3 storage, must exist before running the registrtaion>*

s3. region0=*<region where the bucket is located, refer to list below for applicable values>*

s3.namespace**0**=*<namespace to register the slice store to, separate multiple namespace with comma>*

s3. accesskey0=*<access key of the S3 storage>*

s3. secretaccesskey0=*<secret access key of the S3 storage>*

. . .

s3.id**<n>**=…

s3.bucket**<n>**=…

s3.region**<n>**=…

s3.namespace**<n>**=…

s3.accesskey**<n>**=…

s3.secretaccesskey**<n>**=…

***Applicable values for “region” property:***

* AP\_Singapore
* AP\_Tokyo
* EU\_Ireland
* US\_Standard
* US\_West

***Note:***

* **<n>** refers to the number of the slice store registrations in the file, start from 0.
* To use the *restlet-logging.properties* file, set the JVM property “-Djava.util.logging.config.file” during runtime with the “java” command.

*For sample registration file, refer to s3SliceStores(sample).txt.*

*For sample windows batch file, refer to registerS3FS.bat.*