11-693 Software Method for Biotechnology Homework 3
Name: Tao Guo
id: taog

Report of Execution Architecture with CPE and Deployment Architecture with UIMA-AS

1. Environment Preparation

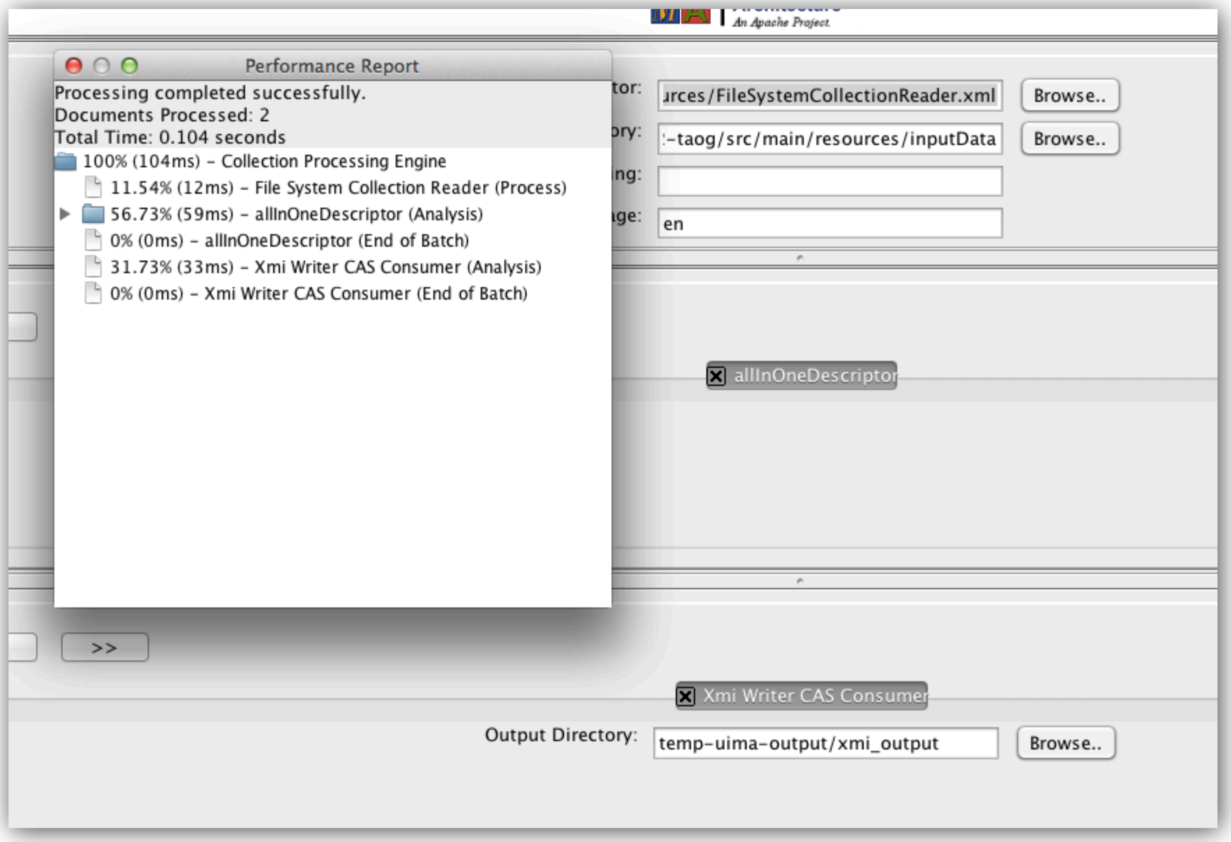
The assignment is finished on Mac OS X 10.8.5. Before the experiment officially start, I have already have Eclipse installed on my Mac. I used the *Install New Software* in the help menu of eclipse to install the newest version of UIMA AS (2.4.0). Do check the environment variable JAVA_HOME, UIMA_HOME and PATH to make sure they work well. If not, just use export in bash to set them.

2. Execution Architecture with CPE

CPE is short for *Collection Processing Engine*, which implements *Collection Processing Architecture*. The *Collection Processing Architecture* helps apply analysis engines to collections of unstructured data, in other words, helps analysis multiple files at the same time.

After learning CPE from the UIMA official documentation and the examples, I started to set up the CPE. As CPE includes an Analysis Engine, which have already been implemented in previous homework, and adds a *Collection Reader*, a *CAS Initializer*, and *CAS Consumer*. The collection reader contains code in the class of *org.apache.uima.tools.components FileSystemCollectionReader* and a *.xml descriptor*, which is implemented as *FileSystemCollectionReader.xml*. The CAS Consumer also contains code in class and descriptor. The class is named *org.apache.uima.examples.xmi.XmiWriterCasConsumer*. and the descriptor is implemented as XmiWriterCasConsumer.

Upon finishing building these components, the CPE will be able to run. Figure 1 shown how it runs.



```
Processing completed successfully.
Documents Processed: 2
Total Time: 0.104 seconds
100% (104ms) - Collection Processing Engine
    11.54% (12ms) - File System Collection Reader (Process)
    56.73% (59ms) - allInOneDescriptor (Analysis)
    0% (0ms) - allInOneDescriptor (End of Batch)
    31.73% (33ms) - Xmi Writer CAS Consumer (Analysis)
    0% (0ms) - Xmi Writer CAS Consumer (End of Batch)
```

urces/FileSystemCollectionReader.xml

:-taog/src/main/resources/inputData

en

Output Directory: temp-uima-output/xmi_output

Figure1: The CPE runs

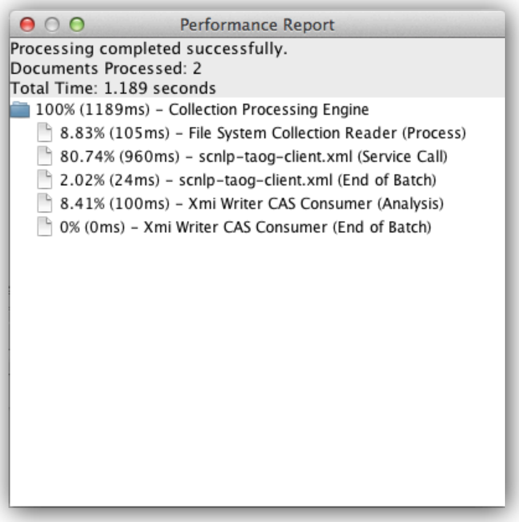
3. Deployment Architecture with UIMA-AS

3.1 Learning UIMA-AS

UIMA-AS is short for UIMA Asynchronous Scaleout. It not only supports collections of unstructured data, but also supports a lot of new features such as multi-threading characteristics. I learned the concepts of UIMA AS from the documentation and samples.

2.2 Creating an UIMA-AS client

In this task, we create a client for the Stanford CoreNLP remote UIMA-AS service and integrate it with the CPE pipeline. I just build an .xml file as scnlp-taog-client.xml setting the broker URL and endpoint. Then set the dependency in the maven project. Figure 2 shows how the client work.



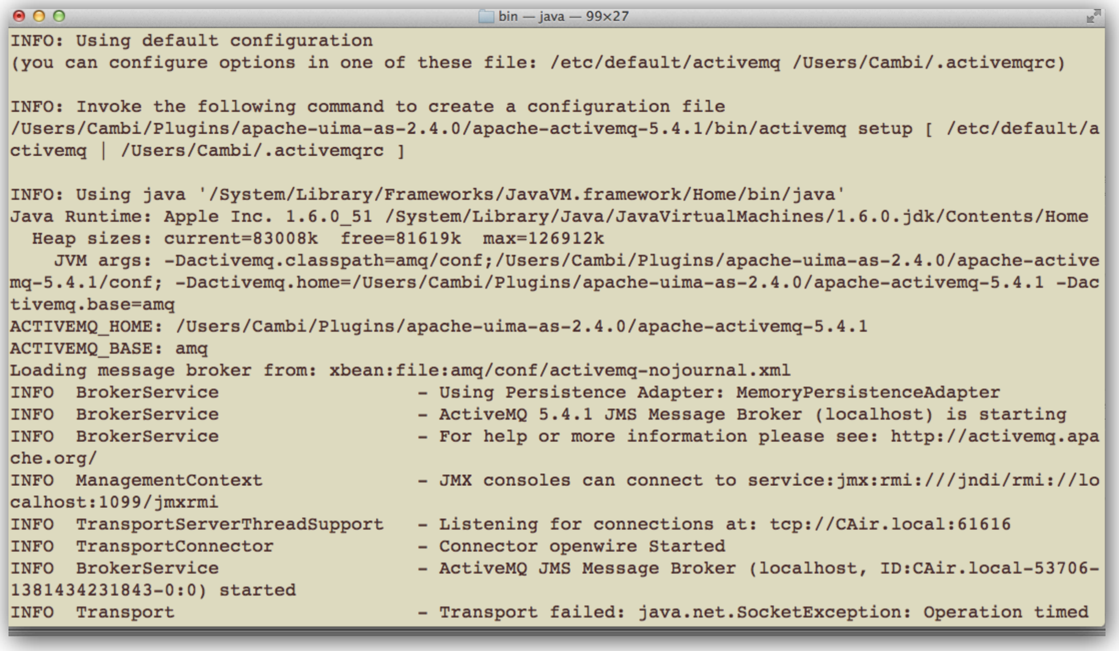
```
Processing completed successfully.
Documents Processed: 2
Total Time: 1.189 seconds
100% (1189ms) - Collection Processing Engine
    8.83% (105ms) - File System Collection Reader (Process)
    80.74% (960ms) - scnlp-taog-client.xml (Service Call)
    2.02% (24ms) - scnlp-taog-client.xml (End of Batch)
    8.41% (100ms) - Xmi Writer CAS Consumer (Analysis)
    0% (0ms) - Xmi Writer CAS Consumer (End of Batch)
```

Figure 2: The Stanford CoreNLP client runs

3.3 Deploying your own UIMA-AS service

In this task, I deployed my aggregate analysis engine in the previous homework on my machine and called the service locally.

I started a UIMA-AS broker locally in the shell. Get the network address and built a deployment descriptor (*hw2-taog-aae-deploy.xml*), then deploy my service to the broker. Figure 3 shows the setting up of the broker.



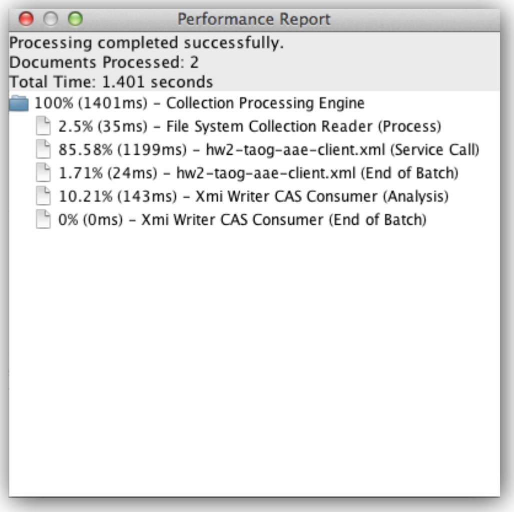
```
INFO: Using default configuration
(you can configure options in one of these file: /etc/default/activemq /Users/Cambi/.activemqrc)

INFO: Invoke the following command to create a configuration file
/Users/Cambi/Plugins/apache-uima-as-2.4.0/apache-activemq-5.4.1/bin/activemq setup [ /etc/default/a
ctivemq | /Users/Cambi/.activemqrc ]

INFO: Using java '/System/Library/Frameworks/JavaVM.framework/Home/bin/java'
Java Runtime: Apple Inc. 1.6.0_51 /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
  Heap sizes: current=83008k  free=81619k  max=126912k
    JVM args: -Dactivemq.classpath=amq/conf;/Users/Cambi/Plugins/apache-uima-as-2.4.0/apache-active
mq-5.4.1/conf; -Dactivemq.home=/Users/Cambi/Plugins/apache-uima-as-2.4.0/apache-activemq-5.4.1 -Dac
tivemq.base=amq
ACTIVEMQ_HOME: /Users/Cambi/Plugins/apache-uima-as-2.4.0/apache-activemq-5.4.1
ACTIVEMQ_BASE: amq
Loading message broker from: xbean:file:amq/conf/activemq-nojournal.xml
INFO  BrokerService                  - Using Persistence Adapter: MemoryPersistenceAdapter
INFO  BrokerService                  - ActiveMQ 5.4.1 JMS Message Broker (localhost) is starting
INFO  BrokerService                  - For help or more information please see: http://activemq.apa
che.org/
INFO  ManagementContext              - JMX consoles can connect to service:jmx:rmi:///jndi/rmi://lo
calhost:1099/jmxrmi
INFO  TransportServerThreadSupport   - Listening for connections at: tcp://CAir.local:61616
INFO  TransportConnector             - Connector openwire Started
INFO  BrokerService                  - ActiveMQ JMS Message Broker (localhost, ID:CAir.local-53706-
1381434231843-0:0) started
INFO  Transport                      - Transport failed: java.net.SocketException: Operation timed
```

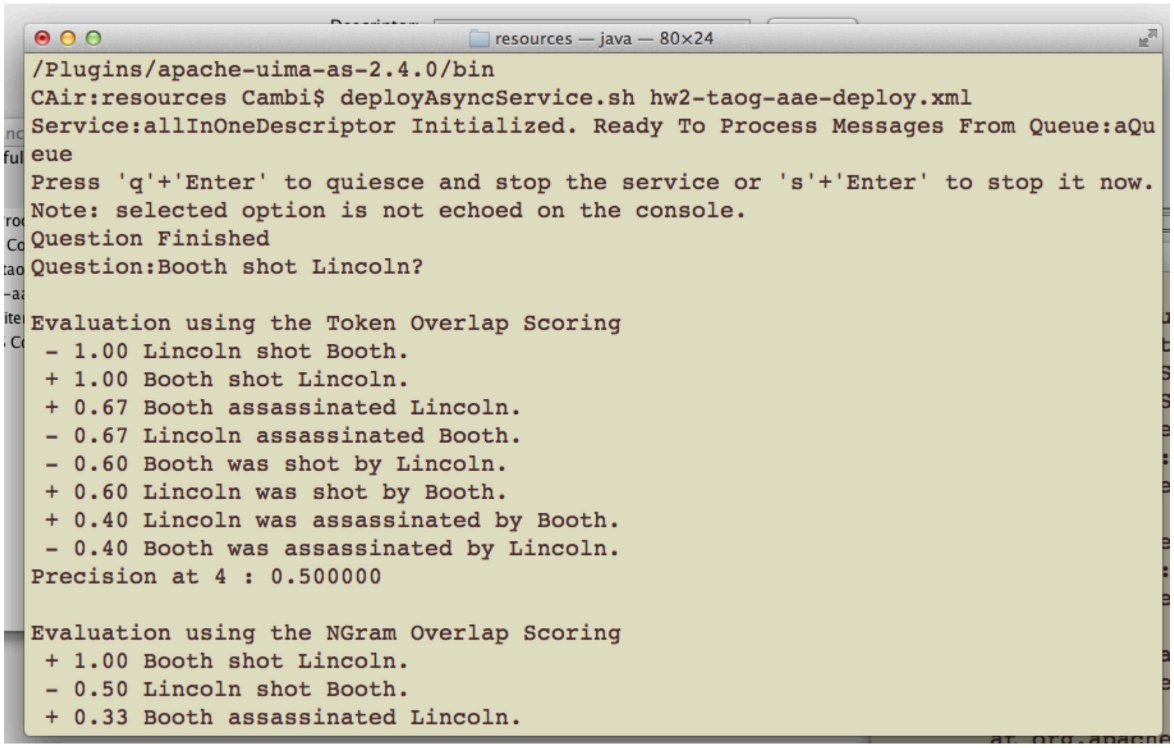
Figure 3. The service is deployed to the broker

Then I build a client (*hw2-taog-aae-client.xml*) in the same way as the client of Stanford CoreNLP service and called the service. The call was a success (Figure 4) and we can get the predict result from the terminal as Figure 5.



```
Processing completed successfully.
Documents Processed: 2
Total Time: 1.401 seconds
100% (1401ms) - Collection Processing Engine
    2.5% (35ms) - File System Collection Reader (Process)
    85.58% (1199ms) - hw2-taog-aae-client.xml (Service Call)
    1.71% (24ms) - hw2-taog-aae-client.xml (End of Batch)
    10.21% (143ms) - Xmi Writer CAS Consumer (Analysis)
    0% (0ms) - Xmi Writer CAS Consumer (End of Batch)
```

Figure 4: The AAE client client runs



```
/Plugins/apache-uima-as-2.4.0/bin
CAir:resources Cambi$ deployAsyncService.sh hw2-taog-aae-deploy.xml
Service:allInOneDescriptor Initialized. Ready To Process Messages From Queue:aQu
eue
Press 'q'+'Enter' to quiesce and stop the service or 's'+'Enter' to stop it now.
Note: selected option is not echoed on the console.
Question Finished
Question:Booth shot Lincoln?

Evaluation using the Token Overlap Scoring
 - 1.00 Lincoln shot Booth.
 + 1.00 Booth shot Lincoln.
 + 0.67 Booth assassinated Lincoln.
 - 0.67 Lincoln assassinated Booth.
 - 0.60 Booth was shot by Lincoln.
 + 0.60 Lincoln was shot by Booth.
 + 0.40 Lincoln was assassinated by Booth.
 - 0.40 Booth was assassinated by Lincoln.
Precision at 4 : 0.500000

Evaluation using the NGram Overlap Scoring
 + 1.00 Booth shot Lincoln.
 - 0.50 Lincoln shot Booth.
 + 0.33 Booth assassinated Lincoln.
```

Figure 5. The predict result shown in terminal