11-791: Design and Engineering of Intelligent Information Systems

Project Individual 8: Deployment

Keith Maki (andrewID: kmaki)

October 26, 2015

1 Introduction

In this report, I present the results of my experiments deploying an analysis engine using UIMA Asynchronous Scaleout. The report is structured as follows: section 2 describes the overall framework pipeline; section 3 provides discussion of important design considerations and limitations to the selected approach; section 4 presents the relevant project files and data; and section 5 concludes the report.

2 Deployment of Analysis Engine

Following the PI8 project description, I deployed an analysis engine using the UIMA AS framework as depicted in Figure 1. I deployed the aggregate analysis engine from the previous Project Individual (aaeDescriptor.xml), which chained two of aggregate analysis engines, one describing a passage ranking system and the other an error analysis framework. For more information on this AAE and its component analysis engines, see the pi7-kmaki report.

3 Design Considerations

The UIMA AS framework makes it easy to host components on servers which are then registered with a broker that serves to interface with client AEs, as seen in Figure 1. By decoupling the analysis engines from the CPE, we allow for much greater degree of asynchronous and parallelizable design. Furthermore, we help to further segment the analysis engine portion of the framework from the CPE, which opens the door to dynamic selection of strategy modules, ultimately enabling systems to learn from their performance as we will explor next week in PI9.

Since the focus this week was on establishing this decoupling and exploring deployment possibilities, rather than refining or improving the performance of the existing analysis framework, the submitted system has not changed significantly from the previous Project

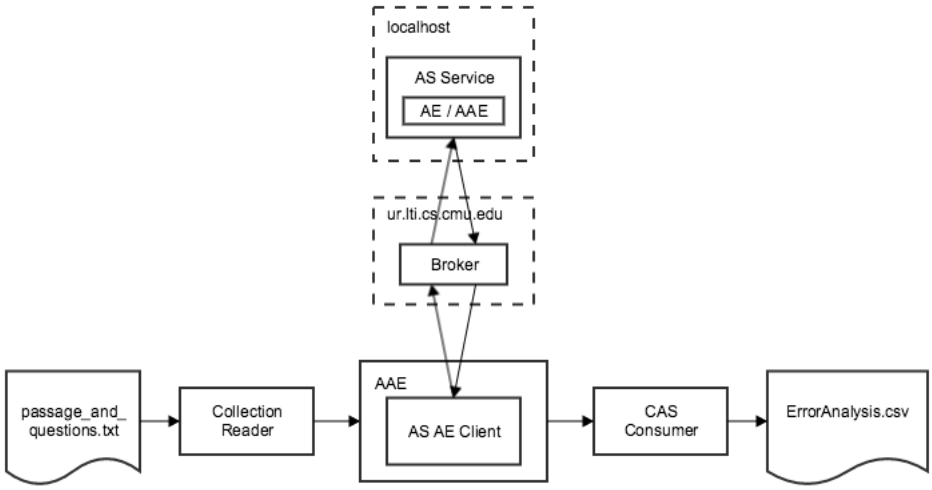

Figure 1: The CPE system architecture of the submitted project

Individual. Please refer to the pi7-kmaki report for a full discussion of design considerations and limitations related to the submitted analysis engines and overall CPE.

4 UIMA AS Deployment Implementation

The full pi8-kmaki project repository including this report can be obtained at https://github.com/kortemaki/pi8-kmaki.

Please note that the structure of the submitted project differs slightly from the suggested project structure outline on TPZ. In particular, the following changes have been made:

- `src/main/resources/deploy -> src/main/resources/descriptors/deploy`
- `src/main/resources/descriptors/ranker_client.xml ->`
  `src/main/resources/descriptors/deploy/kmakiPI8_client.xml`
- `src/main/resources/descriptors/deployRankerAAE.xml ->`
  `src/main/resources/descriptors/deploy/deploy_kmakiPI8.xml`
- `deployRankerAs.xml -> deployKmakiPI8.sh`

To run the project, please first build the project using mvn install, then start the broker server on ur.lti.cs.cmu.edu and run deployKmakiPI8.sh, and finally call mvn exec as usual.

5 Conclusion

In this report, I presented my UIMA Asynchronous Scaleout AAE deployment implementation in pursuit of the requirements of the PI8 assignment. The speed of the resulting deployment system is a testament to the advantage that parallelized and dedicated system designs can bring in terms of simplicity and organization. I found myself surprised that the amount of bandwidth used in transferring the CAS objects over the network was negligable, although it should be expected that the actual analysis engine code should be much more expensive than the bandwidth involved in transferring a CAS. I am looking forward to PI9, and exploring the possibilities presented by this decoupled CPE design in terms of instantiating many AAE instances on remote systems and making decisions about which AAE(s) to use based on system performance.