Abstract

The Diamond project is a client-server distributed system enabling interactive search of large, complex, non-indexed datasets using an early-discard strategy. The traditional Diamond system issues a search request from a single client device to one or more servers, which in turn search over their datasets and stream results back to the client. Mobile Diamond enables Diamond searches from resource-limited mobile devices, which suffer from resource limitations such as lack of memory, poor network connectivity, small display and the inability to compile for or execute code on a certain platform that do not hinder normal client devices.

1 Introduction

Summing up my outline expectations, I estimate the length of the paper to be 8.5-10 pages. This section will be 1 page.

1.1 Diamond

0.5 page. This section will describe the Diamond approach to searching complex data.

1.2 Mobile Scenarios

0.5 page. This section will outline several potentially useful scenarios for interactive mobile search. Ignore the citation here. [1]

2 Background

1.5 pages.

2.1 Mobile Platforms

0.75 pages. Background on existing mobile platforms and the services they provide, such as PalmOS, Symbian, Familiar Linux, Google’s Android, Windows Mobile, etc.

2.2 Mobile Device Limitations

0.75 pages. This section will state the problems that arise when attempting to run a Diamond client on a mobile device as evidenced from the above platforms. They are: (1) the network connection may be low-bandwidth and high latency; (2) the phone may not have enough memory to store search results; (3) the phone may not have a display capable of showing results, or an interface capable of manipulating traditional Diamond applications; (4) the mobile device’s software platform may not allow C code to be compiled and executed.
3 Design and Implementation

4 pages.

3.1 High Level Architectural Overview

1 page, including pictures. Contrast the new architecture with the traditional Diamond architecture, and introduce the new components we will be using.

3.2 OpenDiamond API Division

1 page. Introduce the OpenDiamond library API, which provides core Diamond features, and separate it into those calls relevant to mobile and display clients.

3.3 Display Control Manager

1 page. Describe our meta-service discovery mechanism for finding available resources in the immediate area to function as display clients, and how it tunnels inbetween those clients.

3.4 Wrapped RPC

1 page. Describe the implementation of the idea of wrapping method calls in Java and transporting them across a network link and then executing them in another language (C) using the SWIG package.

3.4.1 SWIG

0.5 page.

3.4.2 Using Java

0.5 page.

4 Evaluation and Results

1 page. I plan to evaluate the overhead involved with making a procedure call across both network and language barriers. This won’t be an attempt to prove any sort of efficiency, but simply a way of showing how slow we can expect it to be, since this is more of an addition of functionality than an increase in performance.

5 Discussion

0.5-0.75 page. I plan to discuss the results of the work.

6 Future Work

1 page. There is a ton of potential future work, such as automating as much of the Wrapped RPC package as possible through stub generation, providing security on the system executing received WRPC calls.

7 Conclusions

0.25 page. Conclude that the project was successful in its attempt to run Diamond on a mobile platform.

References