Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.
Scalable motif search in graphs using distributed computing

Andrew Esler

A thesis presented in partial fulfilment of the requirements for the degree of a Masters in Computer Science

Massey University
Turitea
New Zealand

2012
Acknowledgments

I would like to thank my supervisor Jens Dietrich for the funding and motivation for this project. Patrick Rynhart provided extensive help with Massey’s computing infrastructure. Thanks also to those who were sounding boards for ideas and pointed out my follies.
Abstract

Motif detection allows software engineers to detect antipatterns in software. By decreasing the number of antipattern instances in a piece of software, the overall quality of the software is improved. Current methods to find these antipatterns are slow and return results only when all antipatterns have been found. The GUERY framework is able to perform motif detection using multiple cores and deliver results as they are generated. By scaling GUERY to run on multiple machines, it was hoped that research requiring many queries on a graph could be performed significantly faster than is currently possible.

The objective of this thesis was to research and prototype mechanisms whereby GUERY could be run using a cluster of computers and results delivered as a stream to interested systems. A system capable of running on a cluster of machines and delivering a stream of results as they are computed was developed.
Contents

List of Figures vii
List of Tables ix
List of Listings xi

1 Introduction 1
  1.1 Introduction .................................. 1
  1.2 Project Objective and Scope ................... 5
    1.2.1 Validation .............................. 5
    1.2.2 Changes in the field .................... 6
  1.3 Spikes ........................................ 6
  1.4 Overview of thesis ........................... 7

2 Background 9
  2.1 Graphs and Motifs ........................... 9
  2.2 Graph Query Languages ....................... 14
  2.3 GUERY ........................................ 14
    2.3.1 GUERY Queries ......................... 17
  2.4 Graph Partitioning ........................... 17
    2.4.1 Implementation ........................ 19
  2.5 Cloud Computing ............................. 20
  2.6 Buffering .................................... 21

3 Requirements and Validation Metrics 25
  3.1 Requirements ............................... 25
    3.1.1 Scalability ............................. 27
    3.1.2 Result Streaming ....................... 27
3.1.3 Outside Project Scope .................................. 28
3.2 Validation Metrics ........................................... 29

4 Existing Platforms and Frameworks 31
4.1 BSP ............................................................. 32
4.2 Pregel ........................................................... 33
4.3 Hadoop .......................................................... 34
4.3.1 MapReduce .................................................. 35
4.3.2 Apache Hama .............................................. 37
4.3.3 Giraph ......................................................... 37
4.3.4 GoldenOrb .................................................. 38
4.4 Graph Databases ................................................ 38
4.4.1 Neo4j .......................................................... 38
4.4.2 InfiniteGraph ............................................... 38
4.5 Terracotta ......................................................... 39
4.6 HipG .............................................................. 39
4.7 Other Approaches .............................................. 39
4.7.1 Custom GUERY clustering system ...................... 39
4.8 Summary .......................................................... 40
4.8.1 Comparison Table ......................................... 40
4.8.2 Why Hadoop? ............................................... 41

5 Hadoop GUERY Runner 45
5.1 Design ............................................................ 45
5.1.1 Input to HGR ............................................... 46
5.1.2 Result Streaming .......................................... 47
5.2 Constraints and Limitations ................................. 50
5.3 Local vs Hosted infrastructure .............................. 50
5.3.1 Hardware ..................................................... 52
5.3.2 Amazon Simple Storage Service ....................... 53
5.3.3 Problems Encountered when moving to Amazon ...... 53
5.4 Experimental Evaluation ...................................... 54
5.5 Requirements Validation ..................................... 58
5.5.1 Set up complexity ......................................... 59
5.5.2 Customization required ................................... 59
5.5.3 Find first ...................................................... 59
5.5.4 Find first 10 .................................................. 60
5.5.5 Find all ......................................................... 60
# List of Figures

1.1 *The definition of the subtype knowledge anti-pattern.* 3
1.2 *An instance of the subtype knowledge anti-pattern found in JRE 1.7.* 4

2.1 *A graph with three distinct vertex roles and three distinct edge roles.* 11
2.2 *An example motif with three roles.* 11
2.3 *An example motif with two roles and an edge constraint.* 12
2.4 *Cloud service layers.* 22

4.1 *A diagram showing the layers of abstraction with respect to GUERY.* 32

5.1 *An overview of the HGR system showing how data flows.* 46
5.2 *The path a result takes once it is computed.* 49
5.3 *A graph of job execution time per instance type vs worker count for all three instance types.* 55
5.4 *A graph of job execution time per instance type vs worker count for m1.large and c1.medium instance types.* 56

6.1 *The basic architecture of RabbitMQ.* 72
6.2 *The interaction between DGPROC components.* 75
6.3 *A diagram of the result subscriber notification process.* 76
6.4 *A diagram of how partitions are created and used during job processing.* 81
List of Tables

4.1 A comparison of systems ......................................... 41
# Listings

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>ResultListener Interface</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Subtype knowledge</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Circular Dependency</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>Abstraction without Decoupling</td>
<td>18</td>
</tr>
<tr>
<td>4.1</td>
<td>Word Count Example map function</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>Word Count Example reduce function</td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>Example of a vertex serialised in JSON</td>
<td>48</td>
</tr>
<tr>
<td>5.2</td>
<td>MotifTransformer interface</td>
<td>48</td>
</tr>
<tr>
<td>5.3</td>
<td>MotifProcessingWorker interface</td>
<td>49</td>
</tr>
<tr>
<td>5.4</td>
<td>A possible serialisation abstraction interface</td>
<td>64</td>
</tr>
<tr>
<td>6.1</td>
<td>Multithreaded GQL engine instantiation</td>
<td>67</td>
</tr>
<tr>
<td>6.2</td>
<td>DGPROC GQL engine instantiation</td>
<td>67</td>
</tr>
<tr>
<td>1</td>
<td>The ANTLR grammar file for GUERY queries</td>
<td>97</td>
</tr>
</tbody>
</table>