Using GPUs to Mine Large Scale Software Problem Repositories

Tim Dunn[1,2], Dr. Sean Banerjee[2]

[1]Department of Computer Science, Clarkson University

[2]Department of Electrical and Computer Engineering, Clarkson University

Introduction:

Fixing Flaws in Software

- Large software projects inevitably contain errors and issues
- Users submit "bug reports" describing observed errors
- Developers use these reports to identify and fix problems
- There are too many reports for manual analysis and organization

Bug ID: 915
Summary: (col-align-inherit) implement inheritance of alignment attributes from columns (align, valign, char, charoff, (lang, dir)?)
Description: something about a missing colframe...
Submitter: kipp
Submitted: 9 - 26 - 1998
Modified: 6 - 7 - 2015
Status: NEW
Product: Core
Component: CSS Parsing and Computation
Version: Trunk
Comments: 396 total

Figure 1: An Example Bug Report

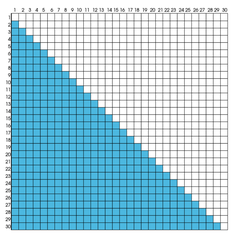
Repository	Total Reports	Average Reports per Day
RedHat	1,357,998	210
Mozilla	1,287,896	197
Novell	989,610	233
Eclipse	498,161	92

Table 1: Bug Repository Sizes

- Several "duplicate reports" often describe the same problem
- Duplicate reports should be identified to increase productivity
- Identifying all duplicate reports requires $O(n^2)$ comparisons
- Even computers require too much time for this computation

$$\sum_{k=0}^{n-1} k = \frac{n(n-1)}{2} \approx \frac{n^2}{2}$$

Figure 2: Report Comparison Complexity



Methodology:

Improving Comparison Efficiency

- Graphics Processing Units (GPUs) are similar to computers, except they have thousands more processors which can operate in parallel
- We utilized a GPU to greatly accelerate problem report comparisons

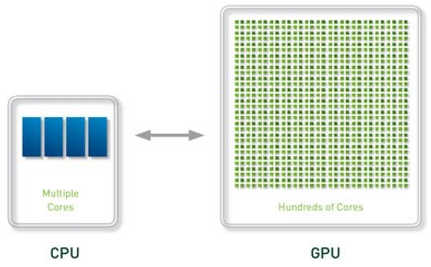


Figure 5: CPU-GPU Comparison[1]

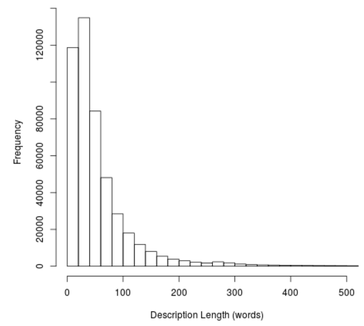


Figure 6: Report Length Distribution

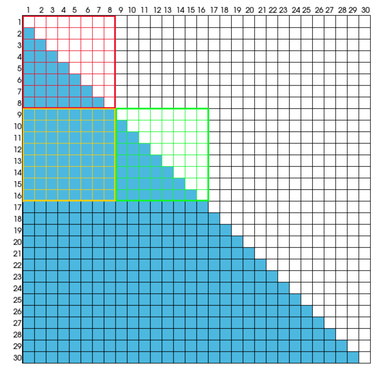


Figure 7: Kernel Tiling

- GPUs are limited by a small memory and the fact that they should operate with arrays of fixed size in order to maximize throughput

Results:

Parallel Algorithms are Faster

- The parallel longest common subsequence and substring algorithms were 86x faster than the serial version
- The cosine similarity algorithm ran 89.8x faster on the GPU

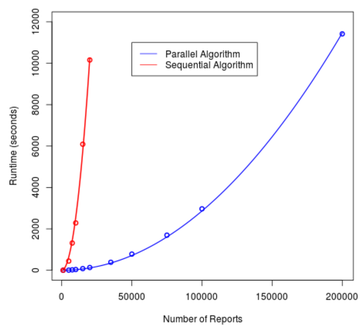


Figure 8: Longest Common Subsequence and Substring Runtimes

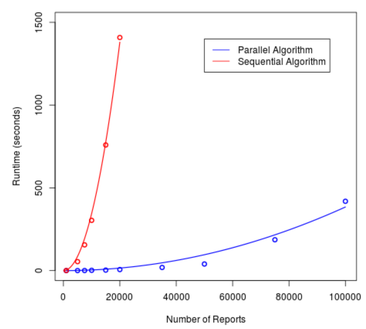


Figure 9: Cosine Similarity Runtimes

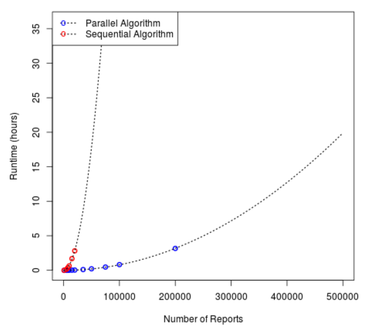


Figure 10: Projected Runtimes

Background:

Current Report Comparison Methods

- Automated approaches find the 20 reports most similar to a new report
- A developer checks each report in the list and decides if it's a duplicate
- Three common methods of report comparison are :
  - Longest Common Substring
  - Longest Common Subsequence
  - The Vector Space Model

Longest Common Substring
right clicking the mouse is not working for me
clicking the help icon does not do anything for several seconds

Longest Common Subsequence
right clicking the mouse is not working for me
clicking the help icon does not do anything for several seconds

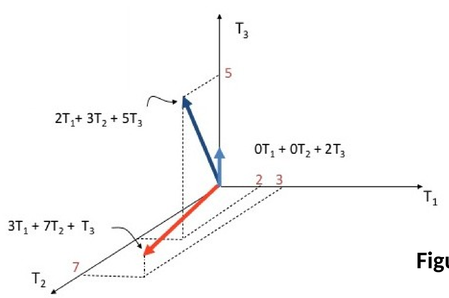


Figure 3: Vector Space Model

NEW REPORT #183

114:	0.257325
29:	0.237171
121:	0.221359
1:	0.217584
168:	0.216930
53:	0.204124
14:	0.202444
18:	0.200000
23:	0.197642
132:	0.195180

Figure 4: Report Similarity Rank

Conclusion:

Knowledge Gained and Future Research

- Parallel report comparison is both possible and practical
- Utilizing GPUs greatly decreases the runtimes of report comparison algorithms

- Several similarity metrics may now be combined to offer more robust comparisons
- Entire datasets may be analyzed, allowing researchers to better test their algorithms
- Combining similarity metrics should retain a high recall rate as repository size increases

Acknowledgements:

I would like to thank:

- Corning Incorporated for the research stipend
- NVIDIA Corporation for the donation of a Tesla K40 GPU
- The Clarkson University Honors Program for everything
- Professor Banerjee for being a great mentor and advisor

References:

[1] http://www.nvidia.com/object/what-is-gpu-computing.html