Chapter 1

My Chosen Application and Kernel

Contributed by My Name

1.1 This is a Dummy Readme Section

Please read this section to understand what to do for your CSE 60742 graph kernel paper, how your "standalone" paper will be integrated into a class **Compendium of Graph Kernels** report that is part of an NSF project, and how at the end you are encouraged to use it as the starting point for an independent publication.

1.1.1 Use as a Test File

The first time a copy of this package is compiled, the following text should be converted correctly:

- This section reference, Section 1.1, is a dummy section to allow you to verify you have imported the template correctly.
- This section reference, Section 1.9., should properly refer to the Sequential Scaling section:
- Citation [1] should be to a paper called "Comparative performance analysis of a Big Data NORA problem on a variety of architectures" and be listed in the bibliography.
- There should be a table labelled Table 1.1 that has characteristics of a Blue Gene computer.
- Fig. 1.1 should properly refer to a figure called "Characteristics of Blue Gene Computers", created from a file"sga-benchmarks.pdf" from directory "figures-xxx".
- Algorithm 1 should be some pseudo-code for an algorithm named "Incremental Jaccard." (This pseudo-code uses the algorithm and algpseudocode packages.)

1.1.2 README

This package, when personalized by a student in CSE 60742, will generate a pdf formatted as a "Chapter." In general, files in the "figures-xxx" directory are the figures for this paper. Files in the "text-xxx" directory are the .tex source latex files for the paper, with each file representing a separate section. Each current file starts with a \section and the name of the section. A label is given for that section in a standard format of "sec:*filename*-xxx". The "refs-xxx.bib" is a .bib (bibtex) file holding all references.

The latex organization of this package will allow the instructor to "easily" compile multiple such student papers into a single "Compendium of Graph Kernels" report, with each student paper a separate chapter, and each having a similar look and feel. However, it is certainly my goal to

Parameter	L	P	Q	Q/P
Cores/node	2	4	16	4X
Core Clock (GHz)	0.7	0.85	1.6	1.9X
Max Node Memory (GB)	1	4	16	4X
Memory Ports per Node	1	2	2	same
Memory B/W per Port	5.6	6.8	21.35	3.1X
Total Memory B/W (GB/s)	5.6	13.6	42.7	3.1X
Inter-node Topology	3D	3D	5D	
Links per Node	12	12	22	4.7X
Bandwidth per Link (GB/s)	0.175	0.425	2	4.7X
Total Link B/W (GB/s)	2.1	5.1	44	8.6X

Table 1.1: BlueGene Family Characteristics.

	Benchmarking Efforts											Outputs						
Kernel	Standalone	Firehose	Graph500	GraphBLAS	Graph Challenge	GAP	HPC Graph Analysis	Kepner & Gilbert	Matenvo	Stinger	Vast 2016	Graph Modification	Compute Vertex Property	Compute Edge Property	Output Global Value	Output O(1) EventsEvents	Output O(|V|) List	Output O(|V|^k) List (k>1)
Anomaly - Fixed Key		S														X		
Anomaly - Unbounded key		S														X		
Anomaly - two-level key		S														X		
BC: Betweeness Centrality						B	B			S			X					
BFS: Breadth First Search			B			B	B	B		B			X				X	
Search for "largest"							B										X	
CC weak: Weakly Connected Component						B		B		S			X					
CC Strong: Strongly Connected Components								B										
Clustering Coefficients										S								
Community Detection										S								
Graph Contraction								B										
GTC: Global Triangle Counting						B									X			
Insert/Delete										S		X						
Jaccard	B/S																	X
MIS: Maximally Independent Set								B										
PageRank						B							X					
SSSP: Single Source Shortest Path						B/S							X				B/S	
All pairs Shortest Path								B										B
TL: Triangle Listing																		B
Geo & Temporal Correlation											S					X		

Figure 1.1: Characteristics of Blue Gene Computers.

Algorithm 1 Incremental Jaccard:
$L,\ R,\ E$ as above
$N(u) = \{w|(G) \text{ in } E\}$

```
1: procedure J2(U, V)
2:     for u in L do
3:         for v > u in L do
4:             γ[u, v] ← 0
5:             for w in N(u) do
6:                 if w in N(v) then
7:                     γ[u, v] += 1
8:             end for
9:         end for
10:    end for
```

encourage students to take their individual "chapters" and adapt them to conference or journal submissions with minimal effort. So please follow the directions below.

1.1.3 Initial Setup Instructions

After down-loading a copy of this package, each student should do the following:

1. Upload the package to a new project in ShareLatex with the name "*xxx-ijk*"where "*xxx*" is a mnemonic or abbreviation for the graph kernel you studied and "*ijk*" is a 3-letter initials of your name. If you are creating multiple chapters, add a number or something to the initials to distinguish between the papers. The goal is to end up with a unique latex project name. An example might be "BFS-PMK".
2. Change the "xxx" in the names for directories "figures-xxx" and "text-xxx", and the file "refs-xxx.bib" to the same kernel abbreviation.
3. In the "text-xxx\body.tex" file, change all "xxx" to the kernel abbreviation.
4. In the "ChapterHeader.tex" file, change the "xxx" in the last line to the kernel abbreviation.
5. In the "text-xxx\redirects.tex" do the following first:
   - Change "My Name" to your name as yo want listed as author.
   - Insert a short 1-word name in the newcommand for kernelName to reflect what kernel you discuss in the paper. It can be the same as your abbreviation.
   - Change the "xxx" in the "*graphicspath*" line to the kernel abbreviation.
   - Insert a longer title as desired in the newcommand for doctitlelong to reflect what this chapter should be called (reflect a key application that would use your kernel and an expanded name for the kernel, such as "Graph Exploration - Breadth First Search").
   - Insert the date of your first version of the paper in the newcommand for originaldate.
6. Check if the whole package still compiles, and look thru the "This is a Sample Section" to see if all references still worked properly.
7. In the "text-xxx\body.tex" file, comment out the "Sample" section with a leading %. (Leaving it in the directory may given you some sample latex formats to copy and paste).
8. Recompile to ensure it still all works. The first section should now be "Introduction".
9. You are free to delete the entries in the .bib file.
10. Please "share" your project with kogge@nd.edu
11. You are now free to add text.

1.1.4 When Filling in Your Own Text

1. PRETEND VERY HARD that what you write will turn eventually into a submission for a conference or journal like KDD. As such, feel free to discuss with your advisor, move sections around, add additional sections as needed, etc. The structure of this latex is to allow you to take the files in the "text-xxx", "figures-xxx", and "refs-xxx.bib" and with little work, plug it into whatever is the main latex template specified for your conference. In particular you should be able to simply insert a reference to the body.tex file into the conference template and have everything port.
2. Also for the class compendium report feel free to add appendices that give more detail.
3. When starting to add text to a section, delete or comment out the paragraph of text I included.
4. Given I expect this paper to evolve over the semester, feel free to comment out from the body file any \input line for a section you are not modifying right now. You can uncomment it when you go back to add text.
5. Place any new text files you generate in the "text-xxx" directory.
6. Feel free to use the Figure and Table sample above as cut and paste as needed.
7. End each label with "-xxx" where "xxx" is your initials. You could also use the "-MyInitials" for this so you don't have to remember your initials. This helps avoid accidentally duplicate labels when multiple chapters are combined.
8. It is not required, but when labelling something, I use a prefix of "sec:" for sections, "fig:" for figures, "tab:" for tables, etc.
9. If you would like to see a table of contents, figures, or tables, uncomment the lines in "main.tex" (remove the "%").
10. Note that I used "[noitemsep,nolistsep,leftmargin=*]" when starting a list to remove both the initial indent and remove extra lins around each item. This is to compact the text as one might do for a page-limited conference paper.
11. The latex packages *algorithm* and *algpseudocode* are great for writing pseudocode.
12. In terms of writing, my graduate advisor insisted on "present tense and active voice" - it results in shorter, clearer, and more readable text. A reference that all of you should have at your elbow is [2]. You can get this from Amazon for under $2,

1.1.5 Paper Development Schedule

You will develop your paper in three phases over the semester.

1. A starting point consisting of a first cut of the Introduction through the Key Graph Kernel, but not including any implementation.

2. An extension covering a reference sequential implementation. This version may be in something as simple as Python.

3. A final extension covering either some sort of an initial parallel implementation or a major revision of the sequential implementation.

At any point, with permission of instructor, if you hit a wall, and want to switch to a different kernel, that can be worked out.

Twice during the semester (after each of the first two passes), each student will both perform two reviews of other papers, and receive three reviews back (two other students and the instructor). This will ensure you learn in detail a bit more about at least four other graph applications and kernels, and get feedback on your own paper that you can use to improve it. Such give and take

will be invaluable as you continue in your graduate research. On each review cycle you will receive a "randomly selected" paper to review, and can choose the other one to review (no repeats and first come first choose when choosing). You will be expected to at least consider each set of reviews in your next revision.

Although actual projects and papers are to be done individually, please feel free to collaborate on graph generators and locating and downloading data sets. If at all possible a common data format such as "'csv" make for an easily interchangeable and processable format.

When you port this to a conference template, delete the file "ChapterHeader.tex, main.tex, and redirects.tex." You can also if you want delete the "sample.tex" file from the "body" directory. You now need only insert a "\input{$text - xxx\body$} into your new template where the body text should go.

1.2 Introduction

Provide an introduction to the problem area(s) for which the kernel you want to study is relevant. For this chapter, you DO NOT need to discuss why graphs are important in general, what graphs are, or other "common" things like how to represent graphs. Such things will be addressed in a common part of the Compendium report. Clearly, though, when you convert your chapter into a paper you will have to add a bit of such stuff back into the intro, and perhaps elsewhere.

1.3 The Problem as a Graph

Discuss here how instances of that problem can be expressed as a graph.

1.4 Some Realistic Data Sets

Discuss here what some data sets that represent instances of the problem might look like and/or are available. What are the number of vertexes or edges for real problems, where are there repositories for such data sets, are there any vertex or edge properties, is there any available graph generators that might be adopted to generating synthetic data sets, and how might such data sets for real applications grow in the future. How does it make sense in your implementation studies to vary the characteristics of test data sets to reflect how real-world data might want to vary?

1.5 Short name for my Graph Kernel-A Key Graph Kernel

Discuss here what a graph kernel that can help solve such problems might be. Pseudo code is fine, but in enough detail that you can project time and space complexity. Discuss also what sort of performance metric makes sense.

1.6 Prior and Related Work

This is space to add in discussion of **prior work** - word on the same problem or kernel that your paper assumes, and **related work** - work on the same application but using different approach or kernel, or a different but similar application..

1.7 A Sequential Algorithm

Discuss here the outlines of a sequential algorithm. What programming paradigms might make the most sense? What are the key data structures? Does the computational complexity differ from that in the Section 1.5?

1.8 A Reference Sequential Implementation

Discuss here your implementation of the basic sequential code. Include what language/paradigm you used for the code.

1.9 Sequential Scaling Results

Discuss here results from your sequential implementation. Include software and hardware configuration, where the input graph data sets came from, and how input data set characteristics were varied. Did the performance as a function of size vary as you predicted?

1.10 A Parallel Algorithm

Discuss here the outlines of a parallel algorithm. Pseudocode is fine. Discuss what you think is the computational complexity.

1.11 A Reference Parallel Implementation

Discuss here an implementation of the basic parallel code. Include what language/paradigm you used for the code.

1.12 Parallel Scaling Results

Discuss here results from parallel algorithm. Include software and hardware configuration, where the input graph data sets came from, and how input data set characteristics were varied. Ideally plots of performance vs BOTH problem size changes AND hardware resources are desired. Did the performance as a function of size vary as you predicted?

1.13 Conclusion

Summarize your paper. Discuss possible future work and/or other options that may make sense.

1.14 Response to Reviews

This will be included only in the second and third iterations, and will be a summary of what you learned from the reviews you received from the prior pass, and how you modified the paper accordingly.

Bibliography

[1] P. M. Kogge and D. A. Bayliss. Comparative performance analysis of a big data nora problem on a variety of architectures. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 22–34, May 2013.

[2] William Strunk and E. B. White. *The Elements of style.* Allyn and Bacon : Longman, Boston (USA), 1999.