

Project Name: Incremental Compilation

Background and Current Studies:

I am a computer engineering graduate at Aligarh Muslim University (2nd Rank in India given by US news and world report) and my broad area of interest are software development, algorithm design and data analysis. The B. Tech course which I am enrolled in has given me a very strong technical hand in these topics. We have courses like software Engineering, Computer graphics, Algorithm Design, Operating System, Computer Graphics, Data base management system. Also we have one software lab in each semester which gives us a chance to apply what we have learned in theory. Apart from the regular curriculum I have an upper hand at basic language like C, C++, Python, Shell, JAVA and Advance java. I have also done a project titled "University student panacea" in which I proposed to replace the time consuming entities such as waiting in queues to fill form and pay application fee which was a major problem at our university to be replaced by digital means. I developed a website which contained the information of all the students of university year wise, their attendance records. Presently I am pursuing a course "Data Analysis" and "Algorithm Design" on COURSERA organization. It is course affiliate to San Diego University which teaches about the basic of data analysis, Algorithm design and HADOOP platform.

Why I wish to participate in Google Summer of Code?

I would gain in depth knowledge about writing scalable and modular code C, C++, Java and Compiler design. I made several mini projects in my college involving language such as C, C++, Java and MYSQL, but none of the project taught me writing code in scalable and modular language. Currently I am learning Compiler design and Theory of Computation (TOC) from various resources on internet and online tutorial from YouTube, NPTEL courses and COURSERA organization. By interning at GSOC (Google Summer of Code) I surely will get a feel of how things are done in the professional world and what's even better I will get to interact with most qualified professionals and students across the globe and learn from them.

Why I wish to work with Chapel project?

While browsing through projects at GSOC I came across your project and it caught my attention for the reason that I have been learning and reading on this topic for a very long time. Now was my chance in front of me to work on a project which was running in my mind from a very long time. The intricacies involved in it like the designing the compiler, writing basic C++ codes to shorten the algorithm are a very interesting prospects of it.

What I do hope to learn over the summer?

Surely working for a big organization like yours would give me an insight about how to work in a team and leadership skills. Given the technical nature of my work I hope to improve my technical skills and learn how to APPLY MY CONCEPTS PRACTICALLY.

Speaking English:

I have professional level English speaking, listening and writing skills.

Do you have any commitments for the summer period? Do you have planned vacation?

No, I have not any planned vacations. I have no obligations during the summer and would be able to contribute full time to GSOC. I will be available full time on GTALK and IRC. I am making it as a primary task in summer and I am willing to work 40 to 48 hours per week.

Name: MD IRFAN

Email-id: irfanmd2503@gmail.com

Mobile-Number: +91-9719821293

IRC Name: irfan_amu

Time Zone: UTC+5:30

Working Hours: I am comfortable to work 10:00-20:00 local time, but since this is my commitment this summer .I can extremely flexible about my working hours.

Permanent Address:

Street: A-73 1st Floor, GALI- TEL MIL, AMAR PURI
NABI KARIM, PAHARGANJ

City: New Delhi

State: Delhi

Country: India

Postal/Zip Code: 110055

Temporary Address:

Street: Room No-115, Top Floor
NADEEM TARIN HALL, Aligarh Muslim University (AMU)

CITY : Aligarh

STATE: Uttar Pradesh

COUNTRY: India

Postal/Zip Code: 202002

Coding Experience:

I have been programming for three years now. I started out with C/C++ in my first year of undergraduate studies and moved ahead to try my hand at JAVA, Advanced java, Python, Shell, PHP, MYSQL. I have also done a mini project titled "University Student panacea" in which I proposed to replace the time consuming entities such as waiting in queues to fill form and pay application fee which was a major problem at our university to be replaced by digital means. I developed a website which contained the information of all the students of university year wise, their attendance records. Apart from these I have also contributed in other mini project with other student project title "Computer Catalogue software" in C++ language.

Survey:

Familiar with tools:

Yes, I am well familiar and already used all tools.

Have you heard about Chapel before the summer of Code? If so where? If not where would you advise us to advertise?

No, I have not heard about Chapel before. I advised to advertise Chapel on Social networking Site such as Facebook, Google+, LinkedIn.

What was the first question concerning Chapel that you could not find an answer to quickly?

Multithreading programming is quite difficult because of race condition, deadlocks and so on my first question is that how does CHAPEL solve them?

What will keep you actively engaged with the Chapel community after this summer is over?

Firstly If I am selected for internship with chapel I will be highly gratified to be a part of it. In return it is the least that I can do to be involved with it actively after the internship period

Prerequisites:

I work at my room on Windows and Linux platform on my personal Laptop. Additionally I use to work in the Laboratory at my department for software /hardware implementation. Both internet access and ability to install new software is not a big problem.

Self-assessment:

What does useful criticism look like from your point of view as a committing student?

I welcome healthy criticism openly as each individual in this world has a different way to look at a problem and I in all my capability may not be able to look with his eyes.

How well can you give constructive advice? How well can you receive advice?

I always try to give the best possible advice using my knowledge whenever asked to do so. If I have no specific idea about the topic being asked, I humbly will opt to answer after a few researches on that topic. Advices from all corners are welcome and if they seem to be proper I will even try to implement them.

What is your development style? Do you prefer to figure out/ discuss changes before you start coding? Or do you prefer to code a proof-of-concept you see how it turns out?

I have a top to down approach of working. I first look at problem as a whole then look at its component and after a careful analysis only start devising a plan to implement it. Then I look at all the possible solutions for implementation and choose the best possible solution. In the process I would love to take feedback from my seniors.

The Task:

Title:

Incremental Compilation

Abstract:

I will introduce the several features that make Chapel more user friendly These includes re-compiling the program only those program parts compile that has been changed, programmed will be automatically saved by reuse the information it already has.

Description Project:

Chapel is an emerging programming language designed for productive parallel computing at scale. It supports a multithreaded execution model via high level of abstraction for data parallelism, task parallelism and nested parallelism.

One of the simple yet effective approaches is to use some hash function and to create a hash table. For every identifier in the table we apply the hash function and calculate some number.

The hash function is an arbitrary function that for each identifier returns some number. It can be a simple sum of the ASCII codes of the identifier or some more complex one. Using hash functions and hash tables is a very effective way to speed up searches in symbol tables. Hash functions and hash tables are used in almost all compilers because their implementation is pretty simple and the gain in search speed is huge.

The compiler operates on an incremental basis, line-by-line, so if only one line is changed in an edit session, then only that line need be recompiled if no other code is affected. Scanning is done incrementally, and the resulting token list saved in memory to be used again where no changes are made. All of the linking tables are saved in code cache memory so there is no need to generate link tables for increments of code where no changes in links are needed. The parser is able to skip lines or blocks of lines of source code which haven't been changed; for this purpose, each line of source text in the editor has a change-tag to indicate whether this line has been changed, and from this change-tag information a clean-lines table is built having a clean-lines indication for each line of source code, indicating how many clean lines follow the present line. All of the source code text modules, the token lists, symbol tables, code tables and related data saved from one compile to another are maintained in virtual memory rather than in files so that speed of operation is enhanced. Also, the object code created is maintained in memory rather than in a file, and executed from this memory image, to reduce delays.

A method of developing source code, comprising the steps of:

- a) editing a module of source text and storing edited source text in a source-text buffer as a plurality of lines, each line having an associated change-tag indicating whether or not the line has been changed since a previous step of editing said module;
- b) compiling said source-text buffer to produce a code table in memory;
- c) linking said code table to produce a code image in a buffer in memory;
- d) running the code via said code image;
- e) step of compiling including scanning a representation of each line of said source text to generate code to build said code table for executable statements of said source text and to generate a symbol table for declarative statements in said source text, each entry in the symbol table corresponding to one of said declarative statements;
- f) step of linking including generating a link table in memory where entries in said link table include an identification and location of an element needed by or supplied by said code table, and including matching said elements needed or supplied in said link table and generating a link

list used in said step of running, and including checking via said change-tag to determine if each of said entries and said link list may be reused;

g) and, after completing or terminating said steps of compiling, linking or running, saving said link table and said link list in memory for reuse another step of editing.

Hence these project focus on the implementation of such functionality in Chapel Compiler, providing such efficient functionality will make Chapel Compiler re-compiling the program only those program parts compile that has been changed by only write the correct syntax of the program and program will be automatically saved.

Benefit:

- Providing such efficiently in Chapel Compiler helps to implement high performance application with minimal programmer effort.
- I will add feature in Chapel Compiler such as recompiling only those parts of the program that has been changed and program will automatically saved by writing just correct syntax of the program and reuse information it already has.

DELIVERABLE:

This project mainly focuses on implementation of Incremental Compilation. I have considered the most useful (Based on my knowledge and mentor suggestion) need to be implementing in this project. At the end of the project the Incremental Compilation with proper documentation in CHPLDOC which comply with Chapel Community standard. The testing and the bug fixing is done after each implemented of project.

I have put some functionality as the optional functionality to implement as they are implemented if in particular time slot after I had done all the functionality which i promise to implement.

Action Plan:

Before April 22nd:

Spent the time on studying the especially Chapel Compiler, Code generation and other Compiler and try to more familiar with the project.

Before May 22nd:

Discuss ideas about introducing Facebook/Twitter/Gmail contact with the mentors. This would make sharing more easily and more content oriented. Discuss with the mentors a testing framework/method to be used for testing the functionalities of the project and spent on studying the code, if other existing code of the related project. It also involves contacting the key people involved in the project to gain the thorough understanding of all the requirement.

After May 22nd:

1. Publish my goals to the Chapel community and try to get some feedback and suggestions on my plans. I will get familiar with the specific Chapel internals that I require for my work (before mid of June).
2. Implement all features in the order they are specified above. That means that I will begin with the implementation of the fast toggle links, continue with live editing and finally build the interactive menu tree. I will try to implement things so that they can be used despite they are not completely finished (until mid of July)

3. Testing of all features with the help of the Chapel community. This will begin as soon as one component is finished and should be completed by the end of July.

5. **Aug 1- Aug 7(7 days):** -----BUFFER TIME 1-----

6. **Aug 8:** ----->Suggested pencil Down---

7. **Aug 9-Aug 15(6 days):**-----BUFFER TIME 2----

8. **Aug 16:**-----Firm pencil Down--

BUFFER TIME:

There is the time reserve 1 week before the suggested pencil down, which serves as the backup time.

OR

Which may spend (as suggest by mentor) on the

- Implementation of the optional functionality which are not implemented.
- Polishing the remaining bit and pieces of the documentation.
- Some additional functionality again as suggested by mentor etc.