TACC
TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

UMass Amherst
The Commonwealth's Flagship Campus

TEXAS
The University of Texas at Austin

Semi-Automatic Code Modernization for Optimal Parallel I/O

SCEC 2018

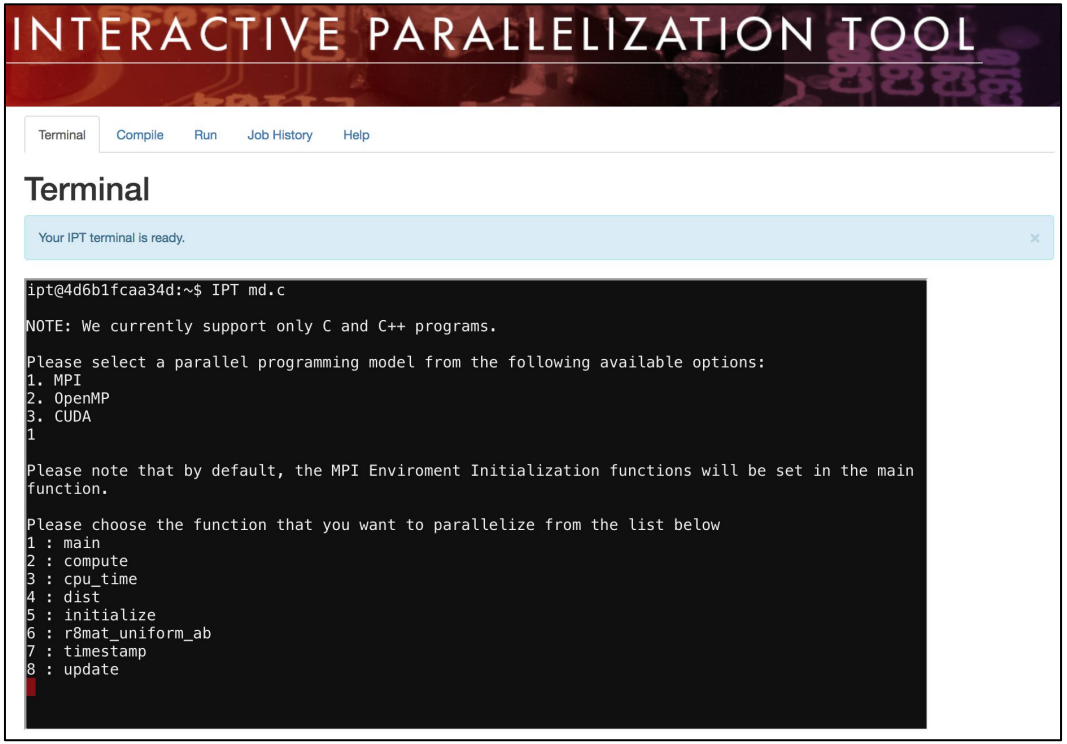
December 14, 2018

PRESENTED BY:

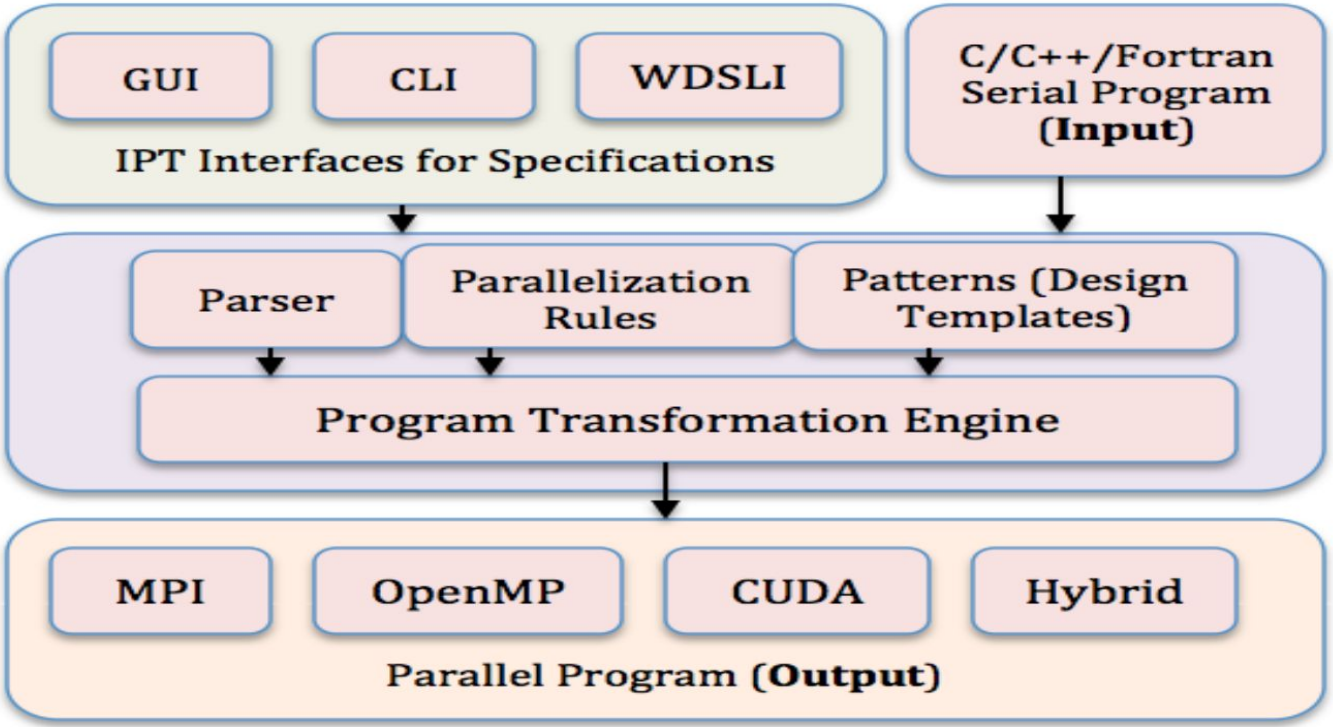
Trung Nguyen Ba: tnguyenba@cs.umass.edu

Ritu Arora: rauta@tacc.utexas.edu

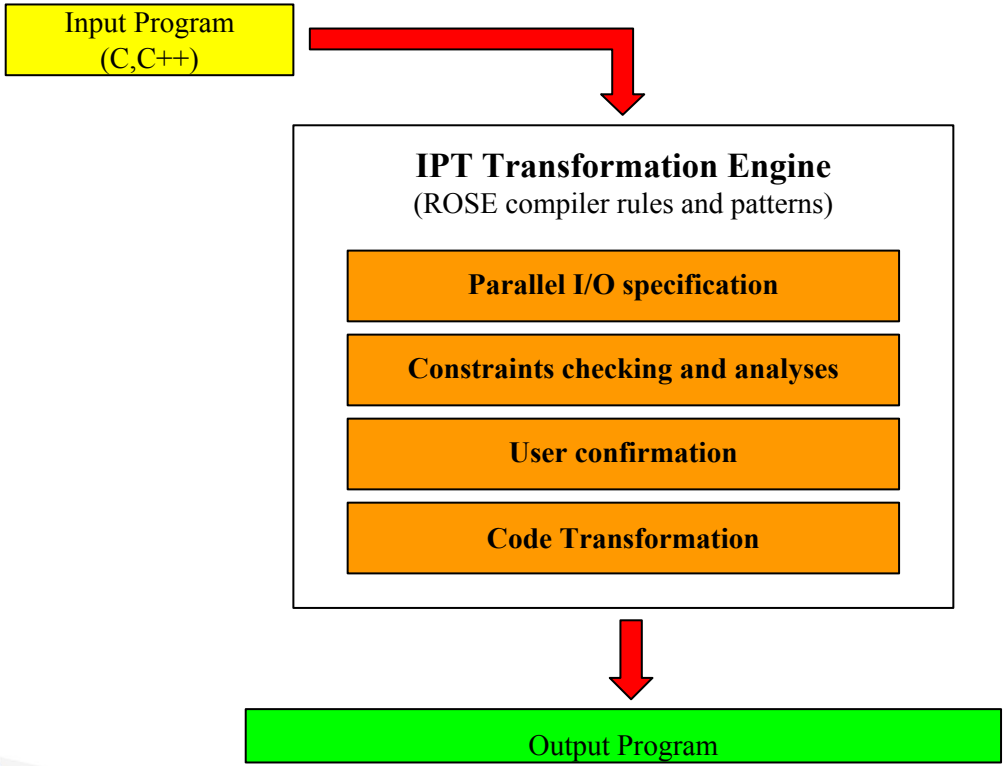
Interactive Parallelization Tool (IPT)



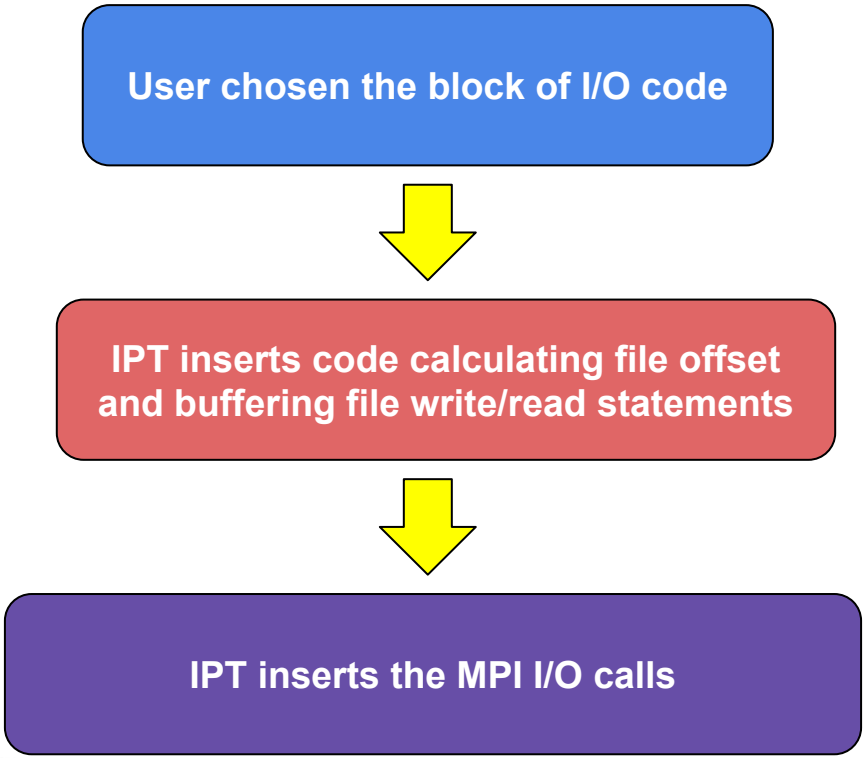
IPT Design Overview



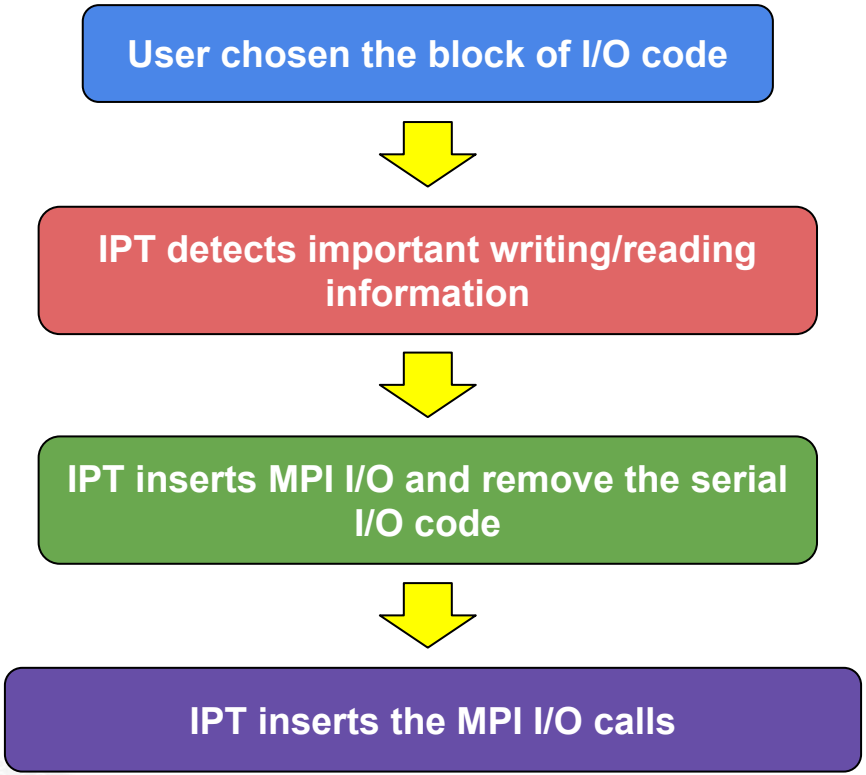
Parallel MPI I/O with IPT



Writing/Reading ASCII Files



Writing/Reading 1-D, 2-D arrays in Binary Files



Example of Optimizable I/O Patterns

Optimizable 1-D array I/O	Optimizable 2-D array I/O

`int a[100];`
`for (int i =0; i < 100;i++) {`
`  fprintf(f,"%d,",a[i]);`
`}`
	`int a[100][100];`
	`for (int i =0; i < 100;i++) {`
	`  for (int j =0; j < 100;j++) {`
	`  fprintf(f,"%d,",a[i]);`
	`  }`
	`}`

Lustre filesystem

- File stripping to increase I/O bandwidth
    - Inserting stripe size
    - Inserting stripe count

Demo

Results and Evaluations

Examples	**Serial Time** Taken in Seconds	**IPT Parallel** Time Taken in Seconds *4 MPI processes used*	**Manual Parallel** Time Taken in Seconds *4 MPI processes used*
1-D Array - reading	42	0.55	0.39
1-D Array - writing	54	1.7	1.66
2-D Array - reading	36	0.53	0.55
2-D Array - writing	40	1.71	1.74

1-D integer array with 100,000,000 elements
2-D integer array with 10,000x10,000 elements

Examples	**Serial** *Total #LoC*	**IPT Parallel** *(#LoC Inserted-or-Deleted) / (#LoC)*	**Manual Parallel** *(#LoC Inserted-or-Deleted) / (Total #LoC)*
1-D Array - reading	11	Lines deleted: 3 Lines added: 32 Total number of lines: 40 %age of code change: 87.5	Lines deleted: 5 Lines added: 16 Total number of lines: 22 %age of code change: 95.5
1-D Array - writing	13	Lines deleted: 3 Lines added: 36 Total number of lines: 46 %age of code change: 84.7	Lines deleted: 6 Lines added: 15 Total number of lines: 22 %age of code change: 95.5
2-D Array - reading	13	Lines deleted: 5 Lines added: 30 Total number of lines: 38 %age of code change: 92.1	Lines deleted: 6 Lines added: 20 Total number of lines: 27 %age of code change: 96.3
2-D Array - writing	18	Lines deleted: 5 Lines added: 38 Total number of lines: 51 %age of code change: 84.3	Lines deleted: 7 Lines added: 24 Total number of lines: 35 %age of code change: 85.6

LoC = Lines of Code

Conclusion

- Overview of parallelizing I/O code with IPT
- IPT supports both ASCII and Binary read and write
  - It also supports file stripping on Luster filesystem
- Performance:
  - IPT-parallel version has almost the same performance as the manual parallel version
  - Reducing the manual effort for parallelizing code for more than 80%

Acknowledgement

The work presented in this paper was made possible through the National Science Foundation (NSF) award number 1642396.