Parallel Knuth Morris Pratt String Matching

CSE 633- Fall 2011

Laurie Dening

Outline

- String Matching
  - Knuth-Morris-Pratt Algorithm
  - How to improve with parallelization
- Experimental Set up
  - Data Description
  - Hardware Specifications
- Results
- Analysis
- Future Work

String Matching

- Goal: Find occurrences of a pattern P of length *m* in a string S of length *n* ($m < n$)

- Applications:
  - Text Processing
  - DNA & Protein sequence matching
  - Anti-Virus & Intrusion Detection
  - Database Query

Knuth-Morris-Pratt

- The prefix function π[q]
  - Encapsulates how the pattern matches against shifts of itself
  - When there is a mismatch, the prefix function tells you how far to shift the pattern

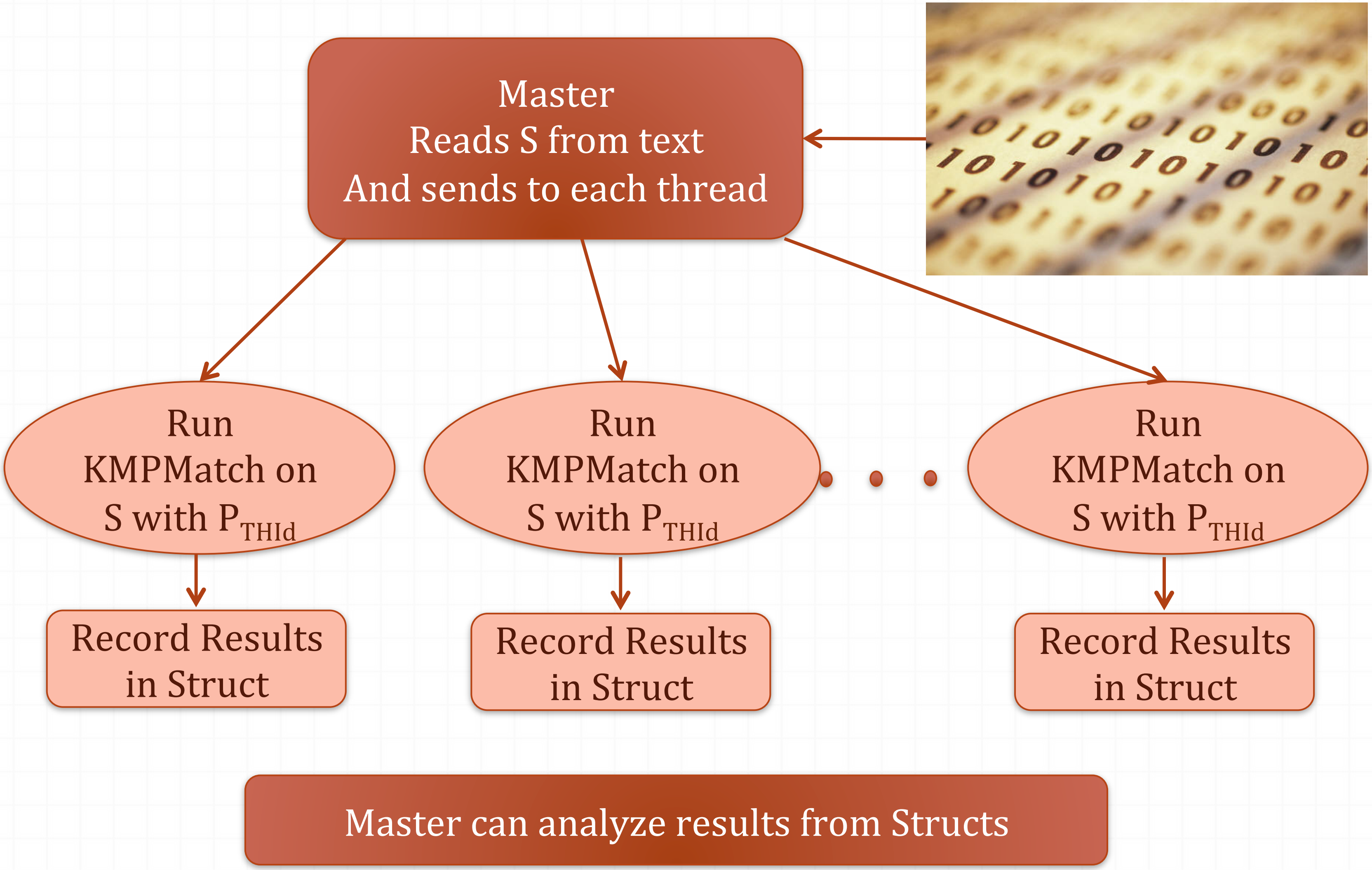
q	1	2	3	4	5	6	7
P[q]	A	B	A	B	A	C	A
Π[q]	0	0	1	2	3	0	1

  - Mismatch: P[q+1] != S[i] -> q = π[q]

In Parallel

- Open MP on Shared Memory Machine
- Matching multiple patterns on same text
- If p = number of patterns, c = number of cores, S = input string
- Find match faster by:
  - Distribute p/c patterns to each core
  - Distribute S to each core
  - Projected speed up approximately c times

Architecture



Experimental Set up - Data

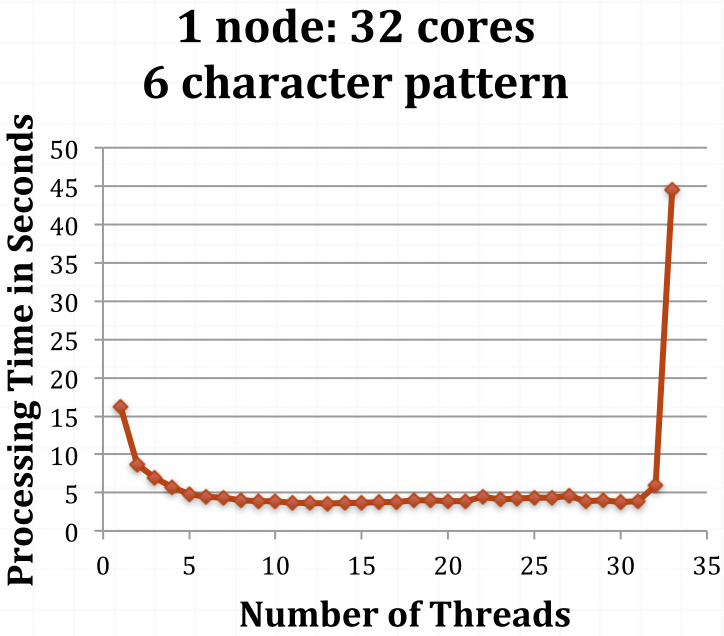
- Input Text
  - 100,000 lines each 256 characters
  - Random 1 and 0
- Pattern
  - Each permutation in order (ie binary counting)
  - 4 'bit' through 13 'bit'
    - 1, 2, and 3 'bit' seemed too trivial
  - Initial trials were fixed on the 6 'bit' pattern
  - Time is average over 3 trials

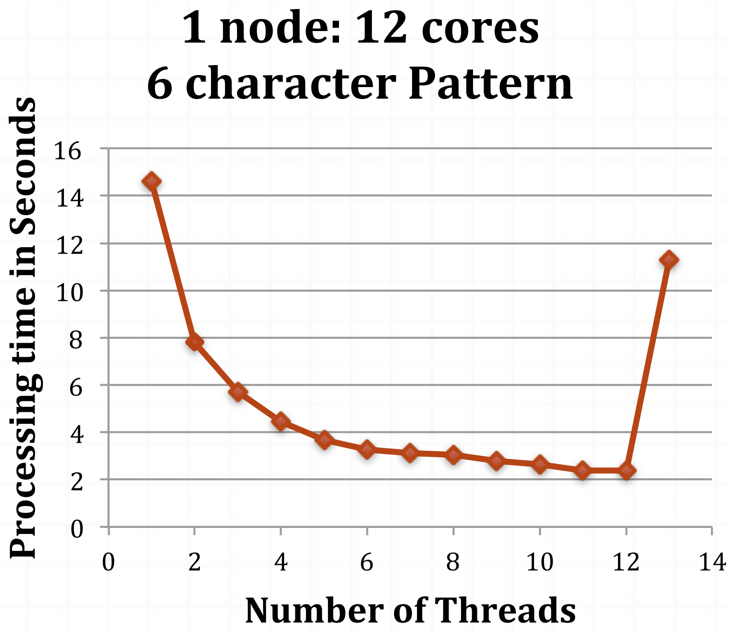
Experimental Set up - Hardware

- 12 Core Machine
  - 2.4 GHz
  - 48GB Memory
- 32 Core Machine
  - 2.13GHz – INTEL
  - 2.2GHz – AMD
  - 256GB Memory

- Initial tests varied threads from 1-64

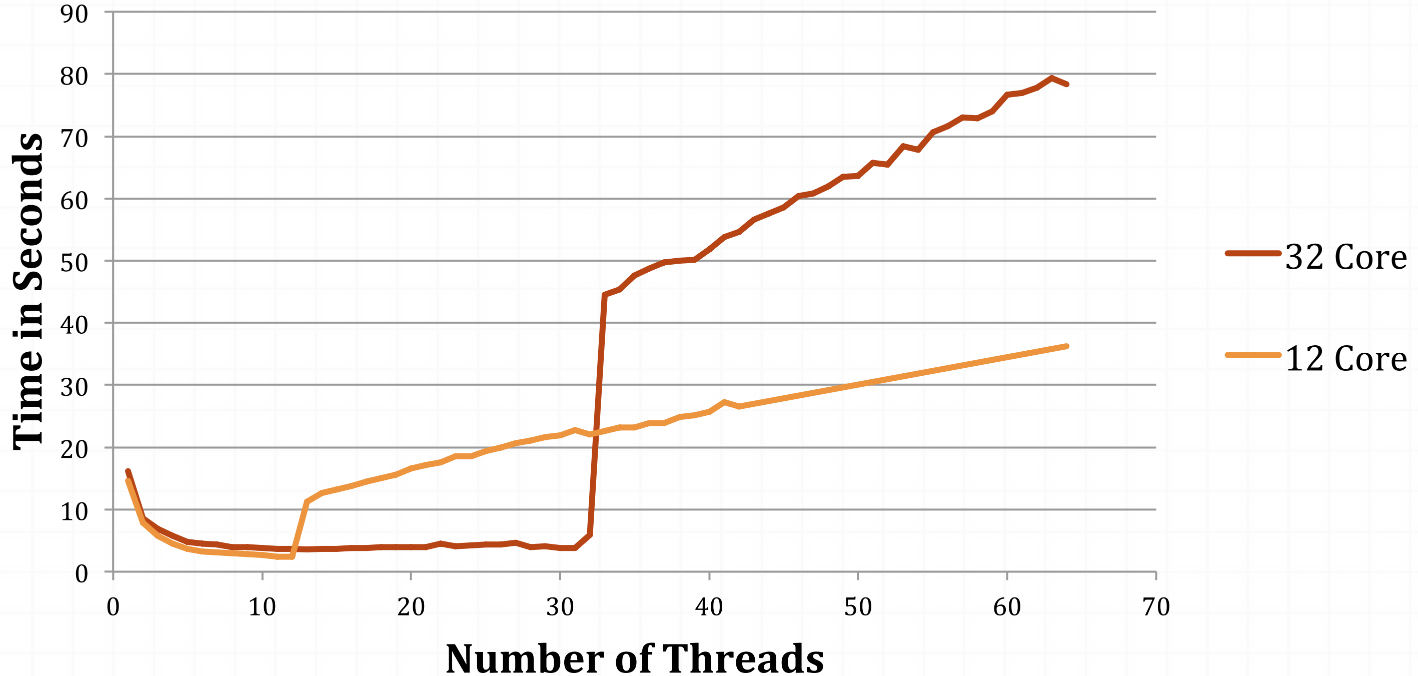
Initial Results



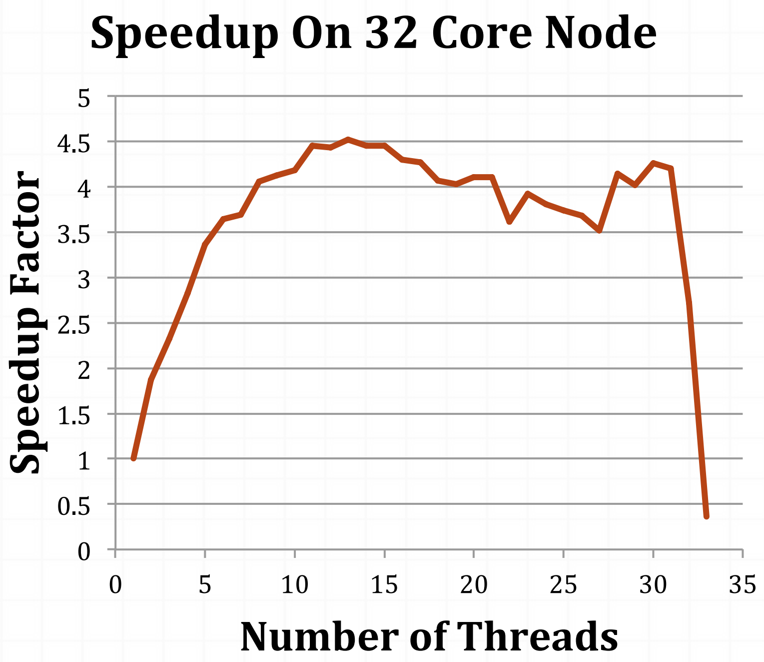


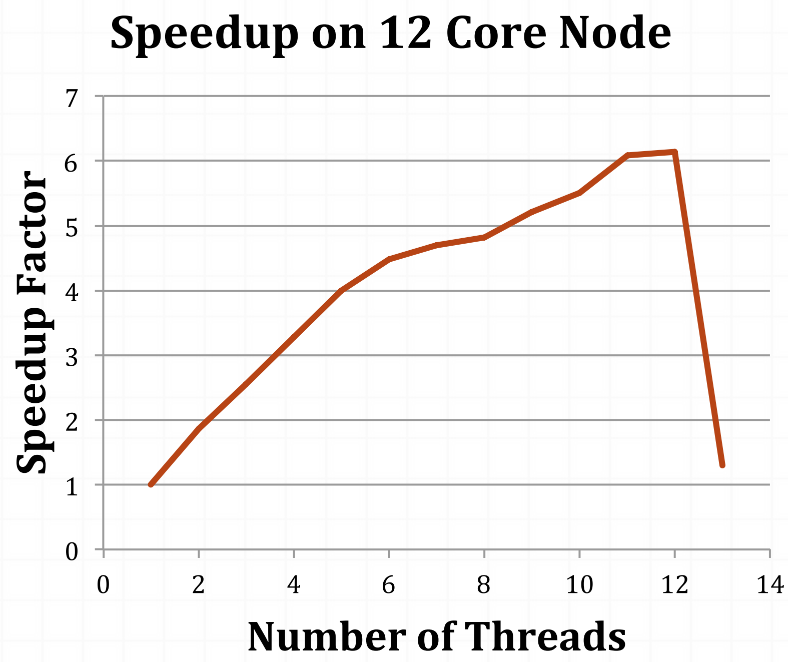
- RAM Time 32 Cores: 16.176 seconds
- Best Time 32 Cores: 13 threads – 3.575 seconds
  - Deviation from 8-21 threads is 0.1575 seconds
- RAM Time 12 Cores: 14.626 seconds
- Best Time 12 Cores: 12 threads – 2.384 seconds

Overall Results for 6 Character Pattern 1 through 64 threads



Speedup

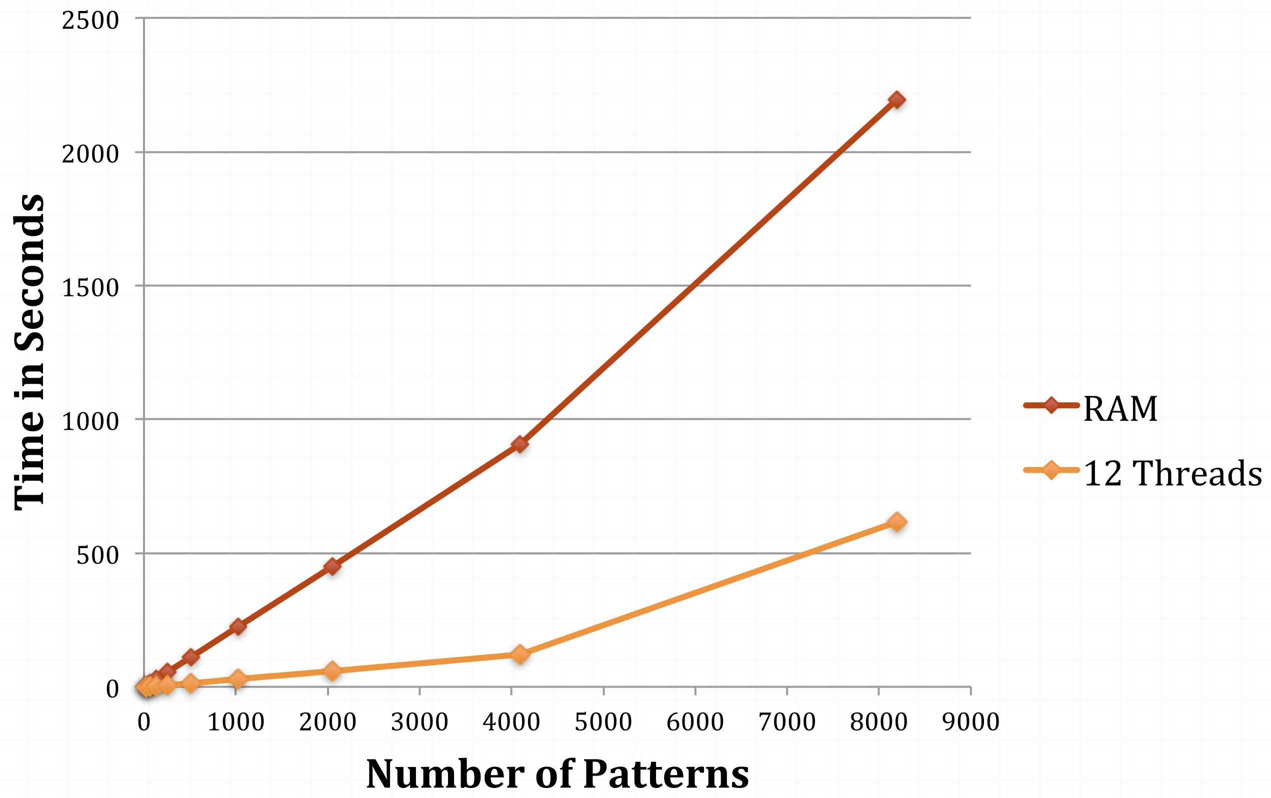


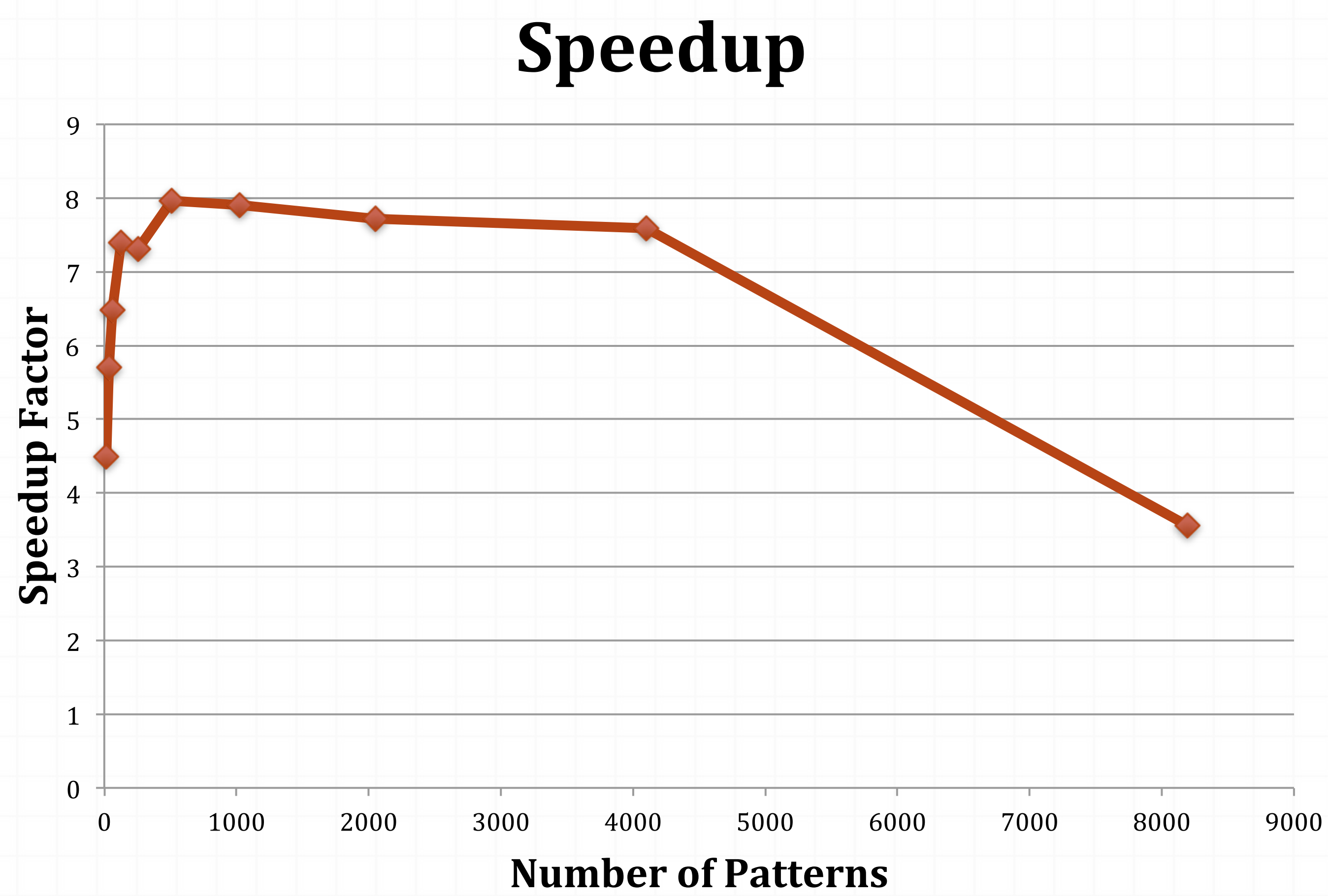


Best Speedup 4.52x

Best Speedup 6.13x

How problem Size effects time




Speedup
Speedup Factor
Number of Patterns
0
1
2
3
4
5
6
7
8
9
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

Summary of Results

- Parallel good until number of threads exceeds number of cores
- 14 ‘bit’, 16k patterns would not run on 12 core machine
- Max speedup on 12 core machine 6.13x not 12x
- Max speedup on 32 core machine even worse 4.52x not 32x!

Future Work

- Only compute prefix function one time for a pattern
- Write results to a file rather than into a struct
  - Memory issues with size of struct per pattern limited the length of pattern
  - Writing to a file solves this issue
- Examine load balancing directives

References

Fast Pattern Matching in Strings Donald E. Knuth, James H. Morris, Jr., and Vaughan R. Pratt, SIAM J. Comput. 6, 323 (1977), DOI:10.1137/0206024

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. "32 String Matching." *Introduction to Algorithms*. Cambridge, MA: MIT, 2009. 1002-006. Print.