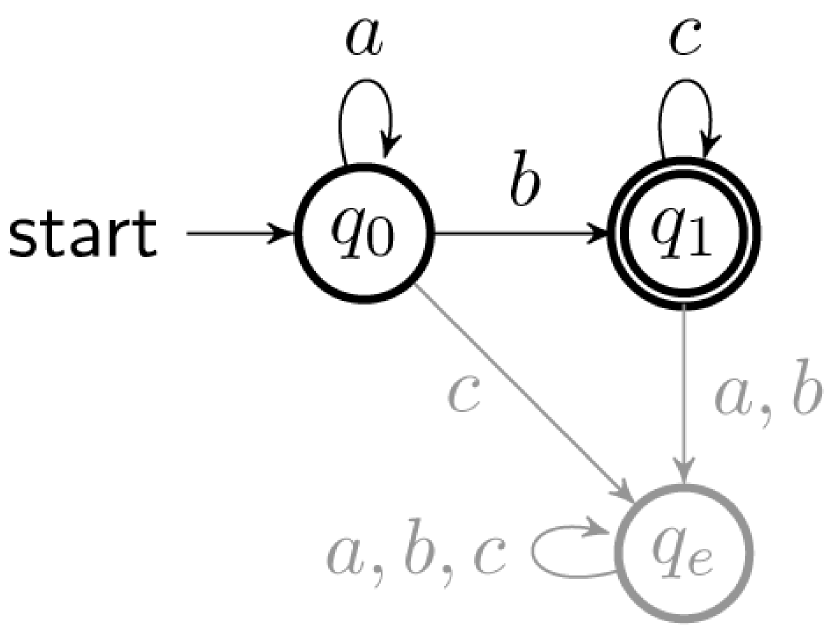
A SPECULATIVE PARALLEL DFA MEMBERSHIP TEST FOR MULTICORE, SIMD AND CLOUD COMPUTING ENVIRONMENT

Yousun Ko
Yonsei University, The University of Sydney

Membership Test

- DFA: Deterministic Finite Automaton
  - Mathematical model of computation to test membership of the regular languages



$$L = a^* b c^*$$

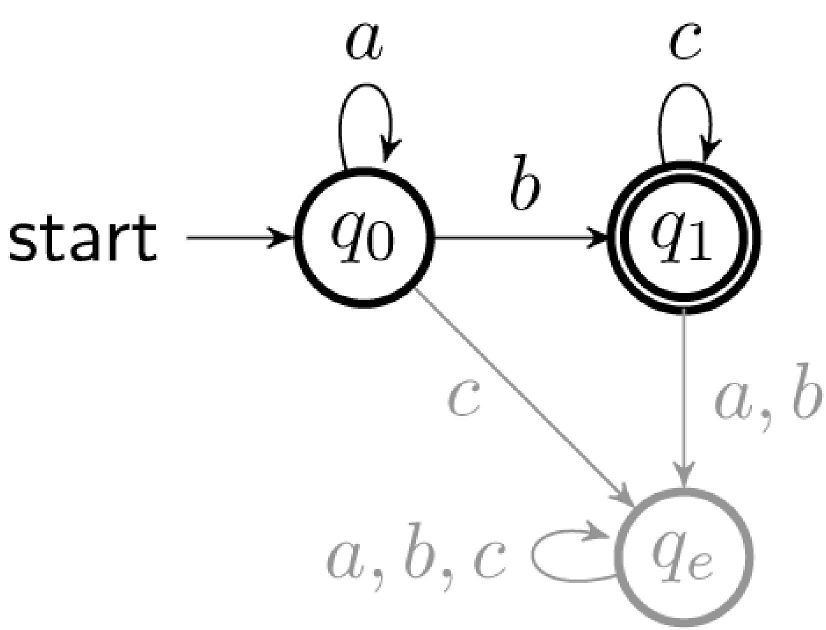
$$M = (Q, \Sigma, \delta, q_0, \{q_1\})$$

$$Q = \{q_0, q_1\}$$

$$\Sigma = \{a, b, c\}$$

Membership Test

- DFA: Deterministic Finite Automaton
  - Mathematical model of computation to test membership of the regular languages



$$L = a^*bc^*$$
$$M = (Q, \Sigma, \delta, q_0, \{q_1\})$$
$$Q = \{q_0, q_1\}$$
$$\Sigma = \{a, b, c\}$$

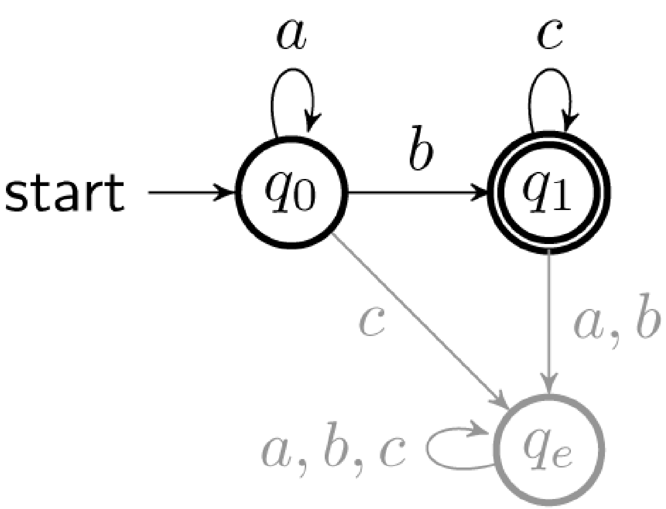
$$Str_0 = aaaaaaabccc$$
$$Str_1 = bbaaaaabccc$$
$$\in L\ ?$$

Applications of DFA

- Text editing, compiler front-ends, web browsers, scripting languages, file-search, command-processors, databases, internet search engines, computer security, DNA sequence analysis aso…

- Why parallelization of DFA membership test is needed?
  - Execution time is originated by transition of states
  - Execution time is proportional to size of input
  - Long input in general
    - 30,000,000,000 long input for DNA sequences analysis

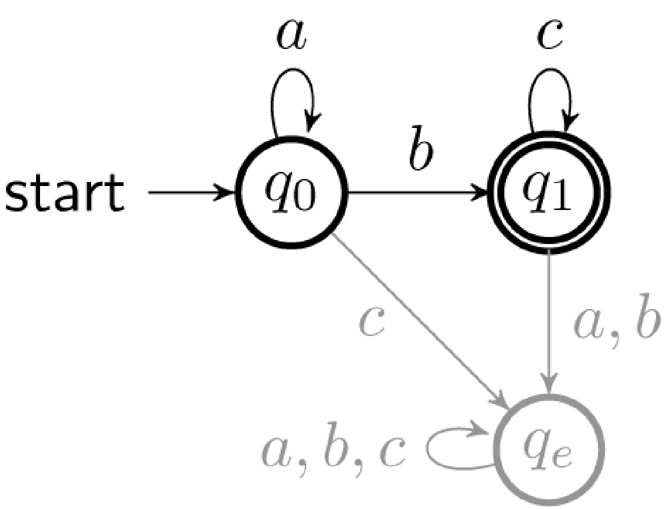
Parallelization of DFA Membership Test



$Str =$ $a\ a\ a\ a\ a\ a\ a\ b\ b\ c\ c\ c$

$p_0 :\ q_0$

Parallelization of DFA Membership Test

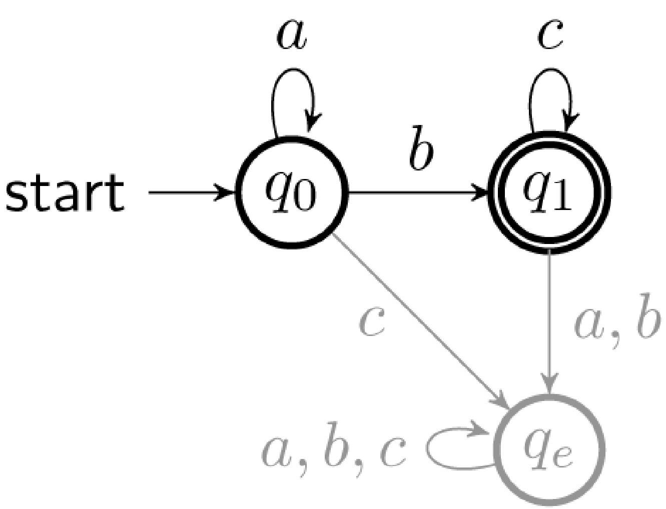


$q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_1\ q_e\ q_e\ q_e\ q_e$

$Str = \boxed{a\ a\ a\ a\ a\ a\ a\ b\ b\ c\ c\ c}$: 12

$p_0 :\ q_0$

Parallelization of DFA Membership Test



$q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_1\ q_e\ q_e\ q_e\ q_e$

$Str = \boxed{a\ a\ a\ a\ a\ a\ a\ b\ b\ c\ c\ c}$: 12

$p_0 :\ q_0$

$Str = \underset{Chk_0}{\boxed{a\ a\ a\ a}}$

$p_0 :\ q_0$

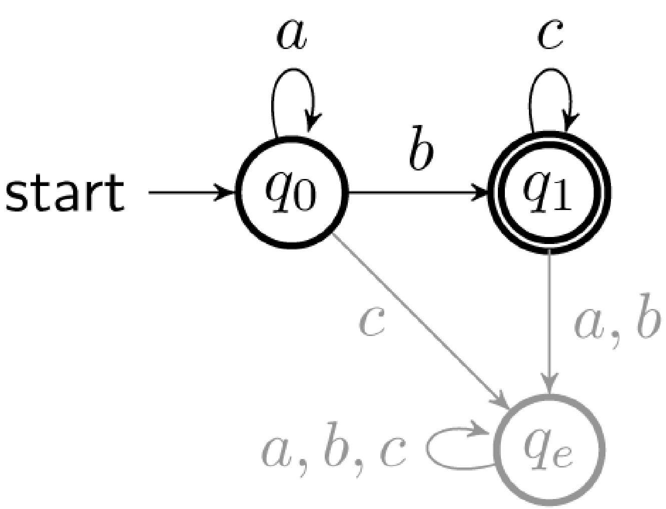
Chk_1: $\boxed{a\ a\ a\ b}$

p_1 : ?

Chk_2: $\boxed{b\ c\ c\ c}$

p_2 : ?

Parallelization of DFA Membership Test



$q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_1\ q_e\ q_e\ q_e\ q_e$

$Str = \boxed{a\ a\ a\ a\ a\ a\ a\ b\ b\ c\ c\ c}$: 12

$p_0 : q_0$

Chk_0

$Str = \boxed{a\ a\ a\ a}$

$p_0 : q_0$

Chk_1

$\boxed{a\ a\ a\ b}$

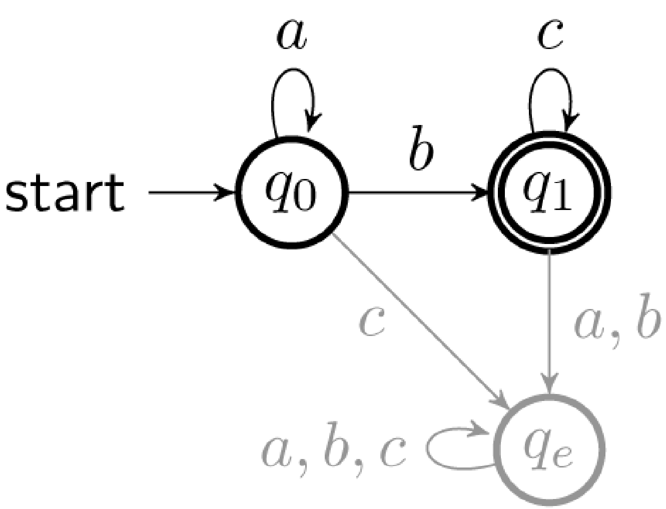
$p_1 : q_0, q_1$

Chk_2

$\boxed{b\ c\ c\ c}$

$p_2 : q_0, q_1$

Parallelization of DFA Membership Test



$Str = \boxed{a\ a\ a\ a\ a\ a\ a\ b\ b\ c\ c\ c}$: 12

$p_0 : q_0$

$q_0\ q_0\ q_0\ q_0\ q_0$: 4

$Str = \boxed{a\ a\ a\ a}$

$p_0 : q_0$

$q_0\ q_0\ q_0\ q_0\ q_0 \parallel q_1\ q_e\ q_e\ q_e\ q_e$: 8

$\boxed{a\ a\ a\ b}$

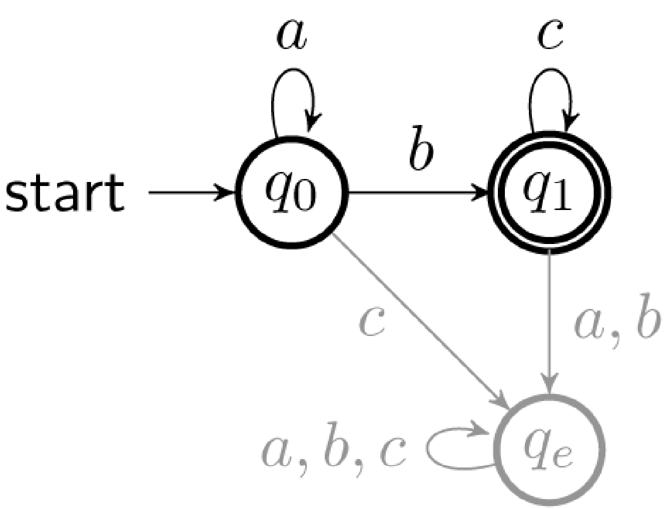
$p_1 : q_0, q_1$

$q_0\ q_1\ q_1\ q_1\ q_1 \parallel q_1\ q_e\ q_e\ q_e\ q_e$: 8

$\boxed{b\ c\ c\ c}$

$p_2 : q_0, q_1$

Parallelization of DFA Membership Test



$Str = $ a a a a a a a b b c c c : 12

$p_0 : q_0$

Chk_0

$Str = $ a a a a a a

$p_0 : q_0$

Chk_1

a b b

$p_1 : q_0, q_1$

Chk_2

c c c

$p_2 : q_0, q_1$

Satisfying:

1) $l_0 = \mathcal{I}\ l_1 = \mathcal{I}\ l_2$ $\quad (\mathcal{I} = |Q| = 2)$

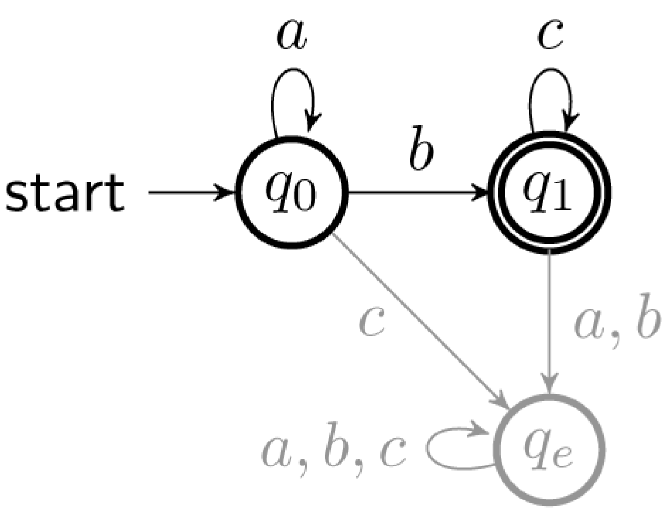
2) $l_0 + l_1 + l_2 = 12$

3) $l_1 = l_2$

$l_0 = 6$

$l_1 = l_2 = 3$

Parallelization of DFA Membership Test



$Str = \boxed{a\ a\ a\ a\ a\ a\ a\ b\ b\ c\ c\ c}$: 12

$p_0 :\ q_0$

$q_0\ q_0\ q_0\ q_0\ q_0\ q_0\ q_0$: 6

$Str = \boxed{a\ a\ a\ a\ a\ a}$

$p_0 :\ q_0$

$q_0\ q_0\ q_1\ q_1 \parallel q_1\ q_e\ q_e\ q_e$: 6

$\boxed{a\ b\ b}$

$p_1 :\ q_0, q_1$

$q_0\ q_e\ q_e\ q_e \parallel q_1\ q_1\ q_1\ q_1$: 6

$\boxed{c\ c\ c}$

$p_2 :\ q_0, q_1$

Satisfying:

1) $l_0 =_{\mathcal{I}} l_1 =_{\mathcal{I}} l_2$ $\quad (\mathcal{I} = |Q| = 2)$

2) $l_0 + l_1 + l_2 = 12$

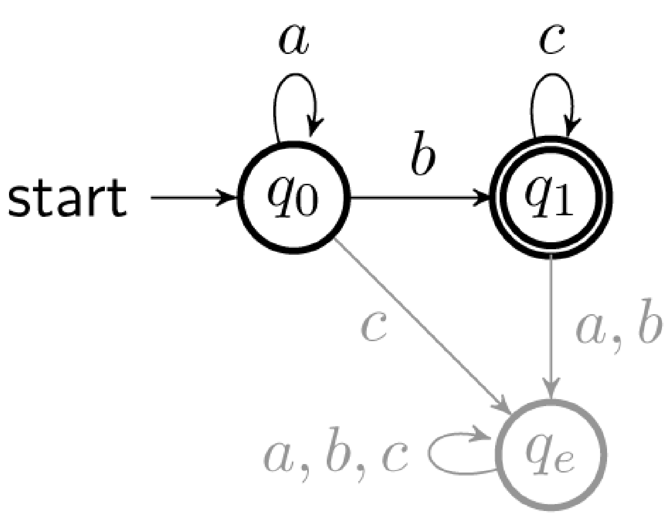
3) $l_1 = l_2$

$l_0 = 6$

$l_1 = l_2 = 3$

Optimization: Reverse Lookahead

- Not all the states are candidates of initial possible states.



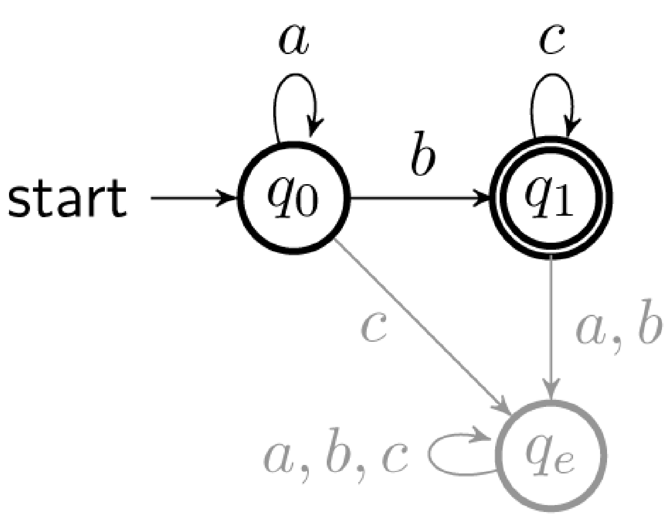
$Str =$ $\underbrace{a\ a\ a\ a\ a\ a}$ (Chk_0) $\quad p_0 : q_0$

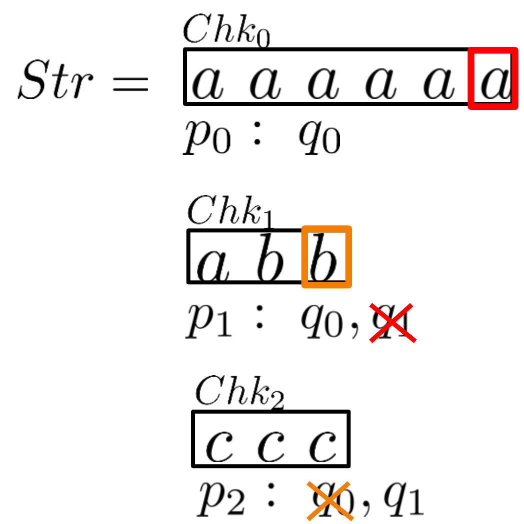
Chk_1: $a\ b\ b \quad p_1 : q_0, q_1$

Chk_2: $c\ c\ c \quad p_2 : q_0, q_1$

Optimization: Reverse Lookahead

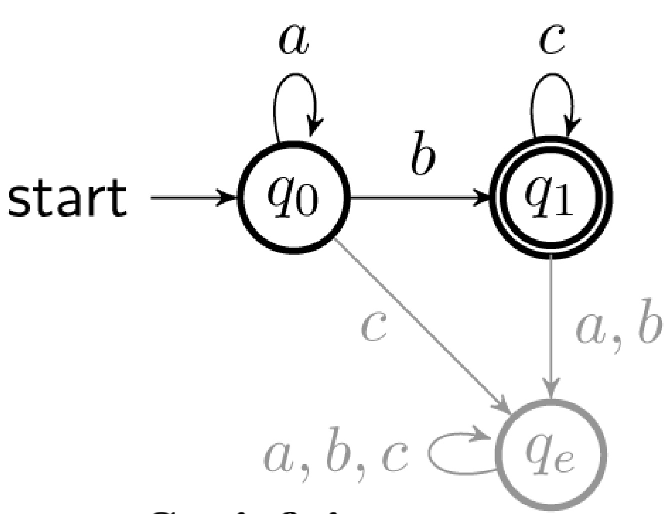
- Not all the states are candidates of initial possible states.





Optimization: Reverse Lookahead

- Not all the states are candidates of initial possible states.
- More reverse lookahead symbols for smaller $\mathcal{I}$



$q_0\ q_0\ q_0\ q_0\ q_0\ :\ 4$

$Str = \boxed{a\ a\ a\ a}$

$p_0 :\ q_0$

$q_0\ q_0\ q_0\ q_0\ q_1\ :\ 4$

$\boxed{a\ a\ a\ b}$

$p_1 :\ q_0$

$q_1\ q_e\ q_e\ q_e\ q_e\ :\ 4$

$\boxed{b\ c\ c\ c}$

$p_2 :\ q_1$

Satisfying:

1) $l_0 =_{\mathcal{I}} l_1 =_{\mathcal{I}} l_2$ $\quad(\mathcal{I} = 1)$

2) $l_0 + l_1 + l_2 = 12$

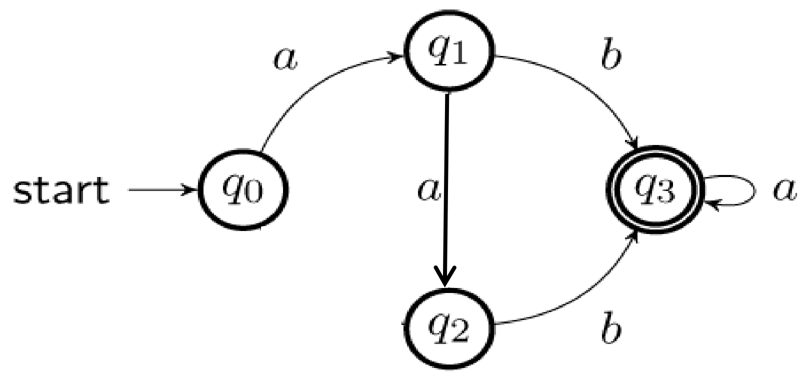
3) $l_1 = l_2$

$l_0 = 4$

$l_1 = l_2 = 4$

Speculation

- What is the **expected** number of possible initial states?



$$\mathcal{S}_a = \{q_1, q_2, q_3\}$$

$$\mathcal{S}_b = \{q_3\}$$

$$\mathcal{I}_{min} \leq \mathcal{I} \leq \mathcal{I}_{max}$$

$$\mathcal{I}_{min} = min(|\mathcal{S}_a|, |\mathcal{S}_b|)$$

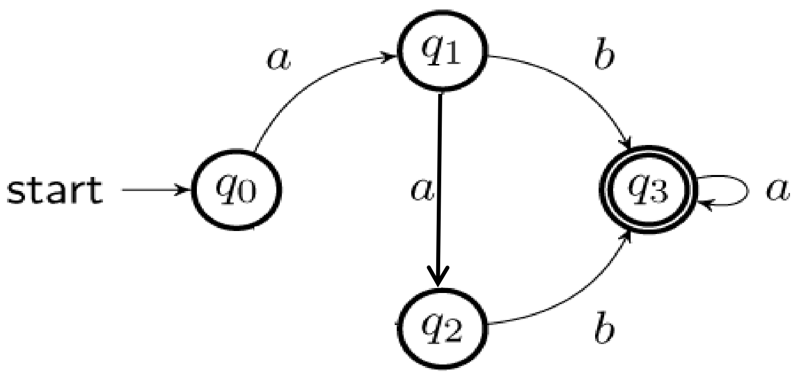
$$\mathcal{I}_{max} = max(|\mathcal{S}_a|, |\mathcal{S}_b|)$$

Optimistic Speculation

Safe Speculation

Speculation

- What is the **expected** number of possible initial states?



$$\begin{array}{l} \color{red}{q_0\ q_1\ q_2\ q_e\ q_e\ q_e\ q_e\ q_e\ q_e} \\ Str_0 = \boxed{a\ a\ a\ b\ b\ a\ a\ a}\ \color{red}{:8} \\ p_0 :\ q_0 \end{array}$$

$\mathcal{I} = |\{q_1, q_2, q_3\}|$

Safe Speculation

$$\begin{array}{l} \color{red}{q_0\ q_1\ q_2\ q_e\ q_e\ q_e\ q_e : 6} \\ Str_0 = \boxed{a\ a\ a\ b\ b\ a} \\ p_0 :\ q_0 \end{array}$$

$$\begin{array}{l} \color{red}{q_1\ q_2\ q_e\ \|\ q_2\ q_e\ q_e\ \|\ q_3\ q_3\ q_3 : 6} \\ \boxed{a\ a} \\ p_1 :\ q_1\ q_2, q_3 \end{array}$$

$\mathcal{I} = |\{q_3\}|$

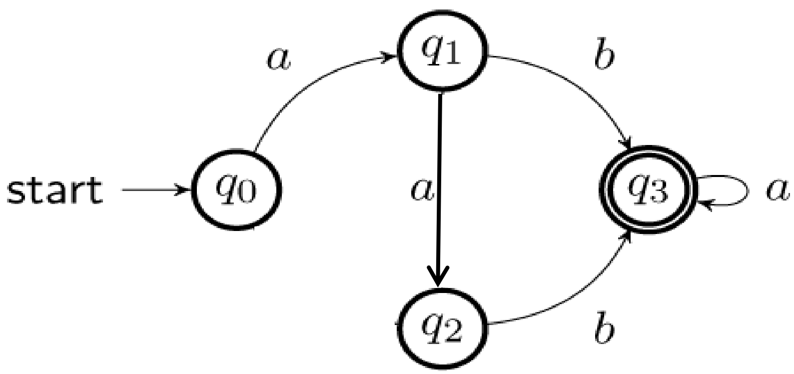
Optimistic Speculation

$$\begin{array}{l} \color{red}{q_0\ q_1\ q_2\ q_e\ q_e\ : 4} \\ Str_0 = \boxed{a\ a\ a\ b} \\ p_0 :\ q_0 \end{array}$$

$$\begin{array}{l} \color{red}{q_3\ q_e\ q_e\ q_e\ q_e\ : 4} \\ \boxed{b\ a\ a\ a} \\ p_1 :\ q_3 \end{array}$$

Speculation

- What is the **expected** number of possible initial states?



$$\begin{matrix} q_0 & q_1 & q_2 & q_e & q_e & q_e & q_e & q_e & q_e \end{matrix}$$
$$Str_1 = \boxed{a\ a\ a\ a\ b\ b\ b\ b} : 8$$
$$p_0 : q_0$$

$$\mathcal{I} = |\{q_1, q_2, q_3\}|$$

Safe Speculation

$$\begin{matrix} q_0 & q_1 & q_2 & q_e & q_e & q_e & q_e \end{matrix} : 6$$
$$Str_1 = \boxed{a\ a\ a\ a\ b\ b}$$
$$p_0 : q_0$$

$$\begin{matrix} q_3 & q_3 & q_3 \end{matrix} : 2$$
$$\boxed{b\ b}$$
$$p_1 : q_3$$

$$\mathcal{I} = |\{q_3\}|$$

Optimistic Speculation

$$\begin{matrix} q_0 & q_1 & q_2 & q_e & q_e \end{matrix} : 4$$
$$Str_1 = \boxed{a\ a\ a\ a}$$
$$p_0 : q_0$$

Failure!

$$q_1\ q_3\ q_e\ q_e\ q_e \parallel q_2\ q_3\ q_e\ q_e\ q_e \parallel q_3\ q_e\ q_e\ q_e\ q_e : 12$$
$$\boxed{b\ b\ b\ b}$$
$$p_1 : q_1, q_2, q_3$$

Merging Local End States

$$\mathcal{L}_i = [l_0, l_1, \ldots, l_{|Q|-1}],$$

where $0 \leq i < |P|$ and $l_j \in Q$ for all $0 \leq j < |Q|$

$q_0 \; q_0 \; q_0 \; q_0 \; q_0 \; q_0 \; q_0$

$Str = \boxed{a \; a \; a \; a \; a \; a}$

$p_0 : \; q_0$

$\mathcal{L}_0 = [0, \phi, \phi]$

$q_0 \; q_0 \; q_1 \; q_1 \quad || \quad q_1 \; q_1 \; q_0 \; q_0$

$\boxed{a \; b \; b}$

$p_1 : \; q_0, q_1$

$\mathcal{L}_1 = [1, 0, \phi]$

$q_0 \; q_e \; q_e \; q_e \quad || \quad q_1 \; q_1 \; q_1 \; q_0$

$\boxed{c \; c \; c}$

$p_2 : \; q_0, q_1$

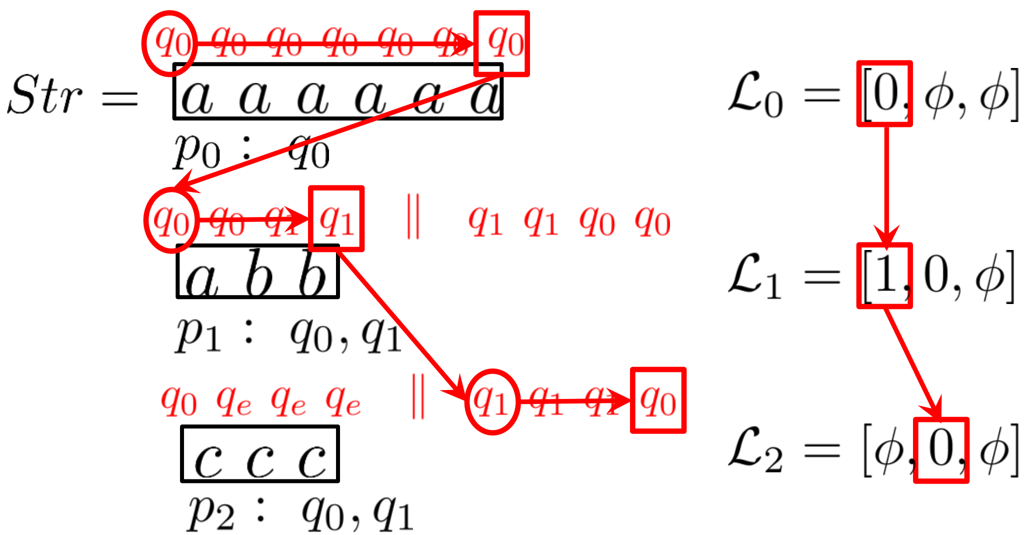
$\mathcal{L}_2 = [\phi, 0, \phi]$

Merging Local End States

$$\mathcal{L}_i = [l_0, l_1, \ldots, l_{|Q|-1}],$$

where $0 \leq i < |P|$ and $l_j \in Q$ for all $0 \leq j < |Q|$

- Sequential Merging

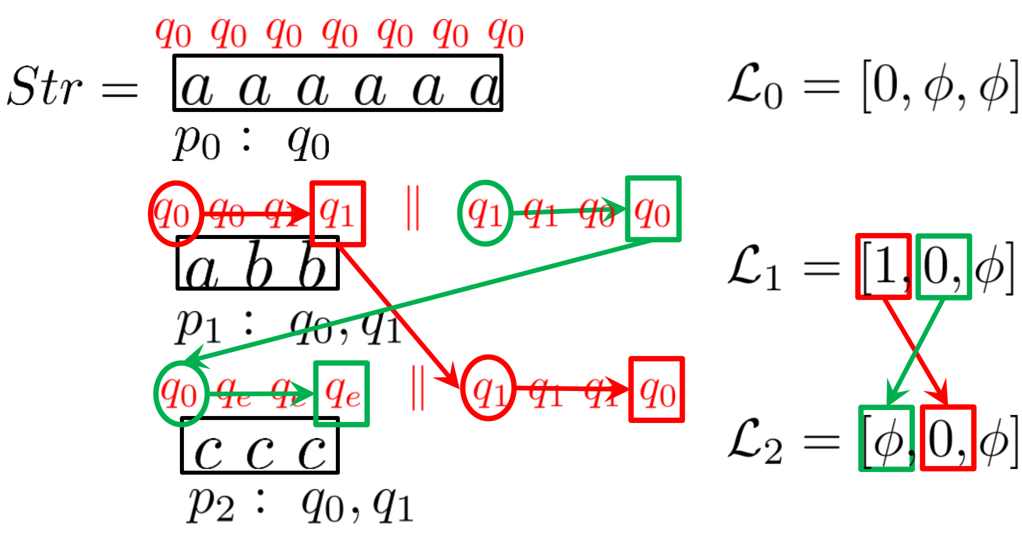


Merging Local End States

$$\mathcal{L}_i = [l_0, l_1, \ldots, l_{|Q|-1}],$$

where $0 \leq i < |P|$ and $l_j \in Q$ for all $0 \leq j < |Q|$

- Parallel Merging



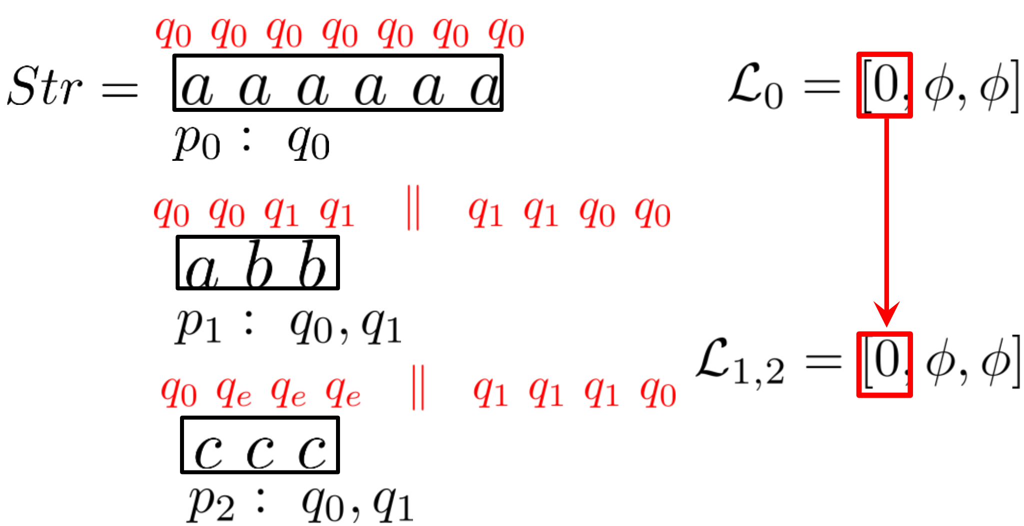
$$\mathcal{L}_{1,2} = [0, \phi, \phi]$$

Merging Local End States

$$\mathcal{L}_i = [l_0, l_1, \ldots, l_{|Q|-1}],$$

where $0 \leq i < |P|$ and $l_j \in Q$ for all $0 \leq j < |Q|$

- Parallel Merging



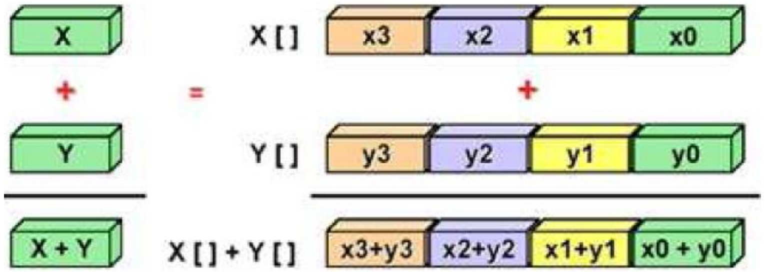
Time Complexity

$$\mathcal{O}(\frac{nm}{m+p}),$$

where $n = |Str|$, $p = |P|$ and m is either $|Q|$ or $\mathcal{I}$.

Experiments: H/W Overview

- Shared-memory multicore
  - Intel Manycore Testing Lab (MTL) : 4CPUs × 10 cores/CPU = 40 cores
- SIMD (Single Instruction Multiple Data)
  - AVX2 instruction set extension (8-fold vectorization)

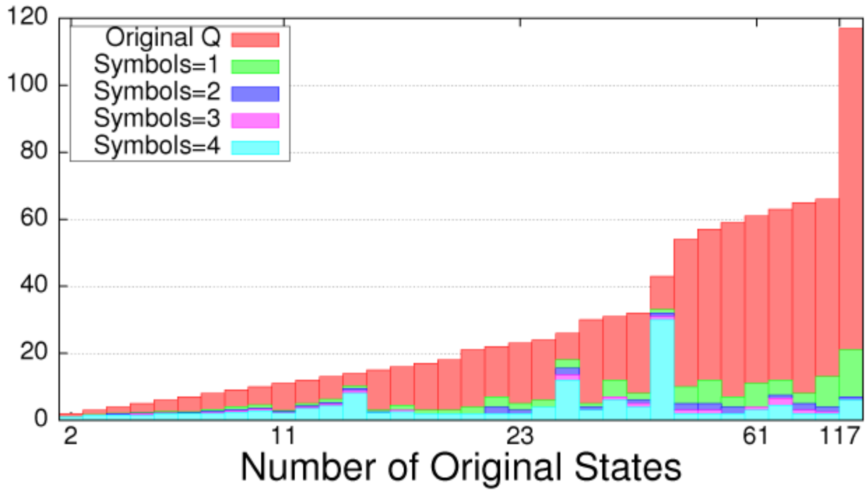


- Cluster Computing Environment
  - Amazon Elastic Cluster Computing (EC2) Environment
  - 288 cores

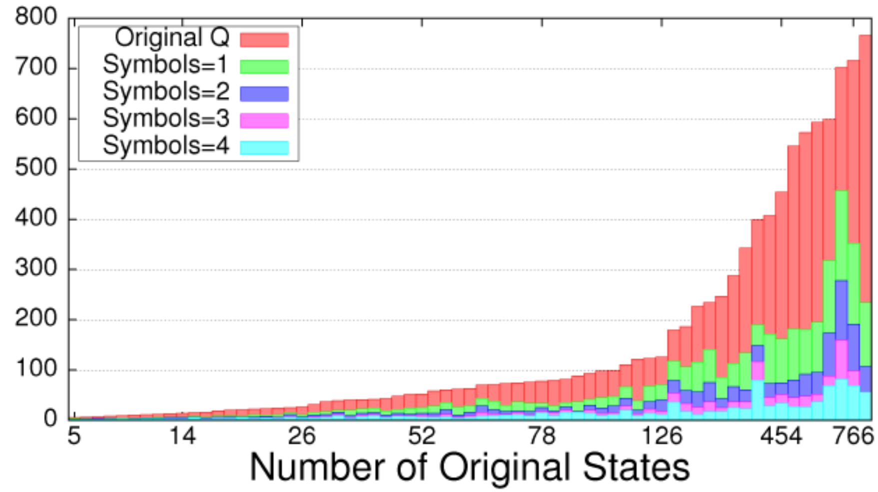
Experiments: Benchmarks

- 299 PCRE (Perl-compatible Regular Expression) RE patterns
- 110 PROSITE protein patterns
- 1MB long input
  - Longer input results better performance!

Reverse Lookahead Symbols



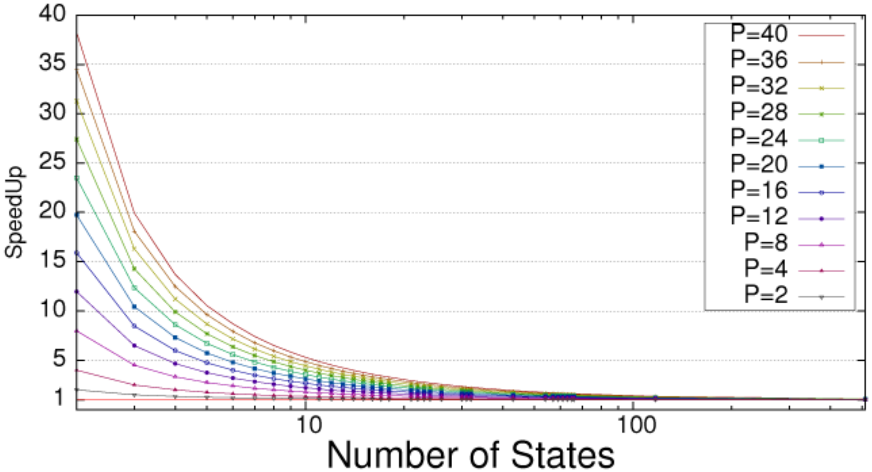
(a) PCRE



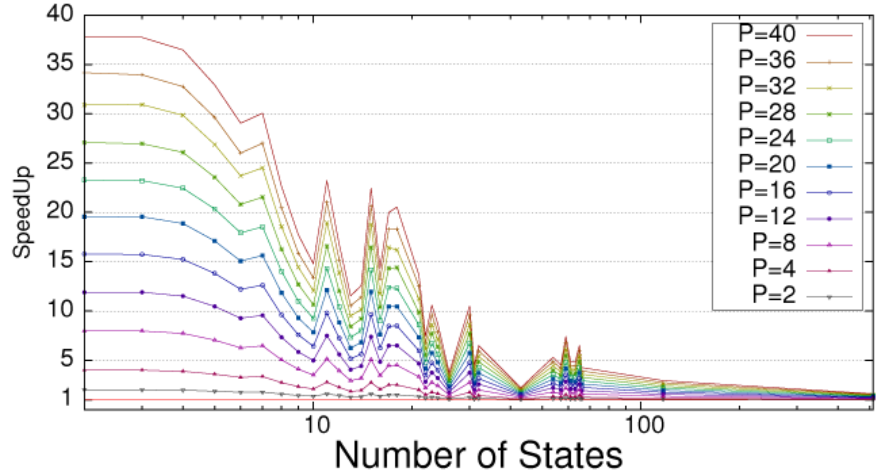
(b) PROSITE

r	0	1	2	3	4
PCRE	100%	33.7%	26.4%	23.7%	21.7%
PROSITE	100%	47.2%	29.2%	20.5%	16.0%

Shared-Memory Multicore

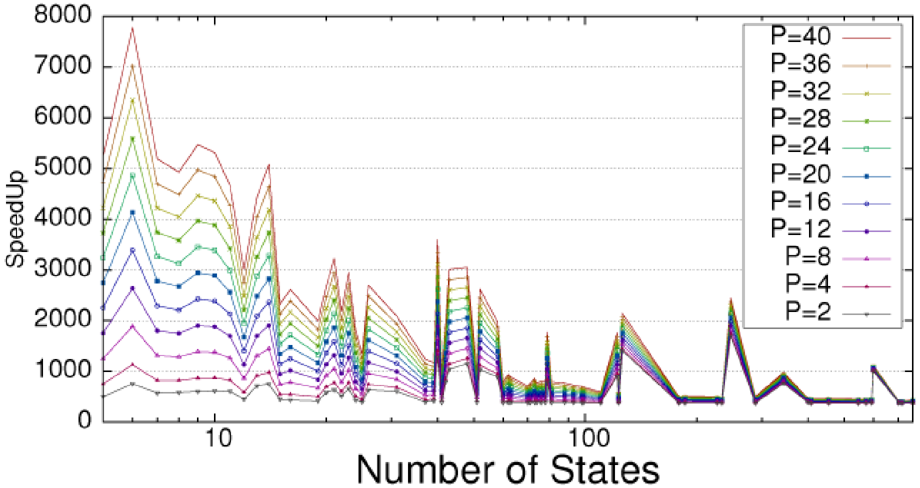


(a) Speedup without optimization for PCRE

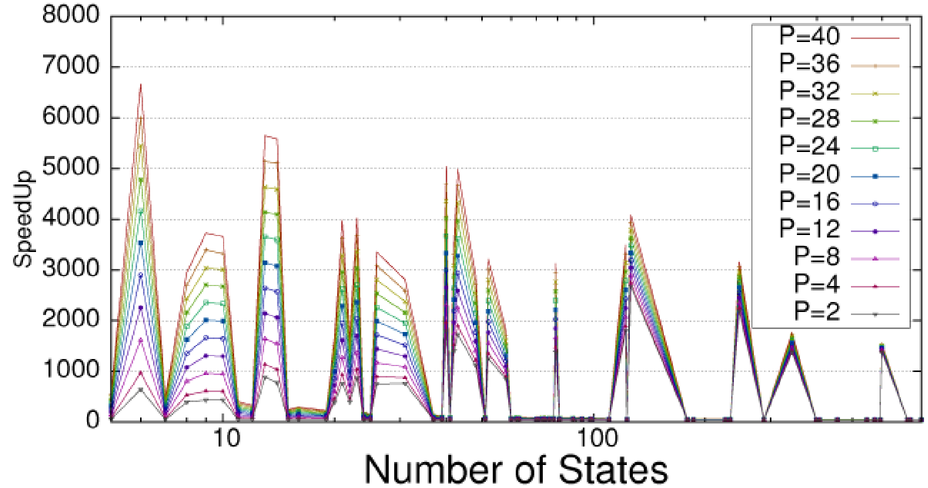


(b) Speedup with optimization for PCRE

Shared-Memory Multicore

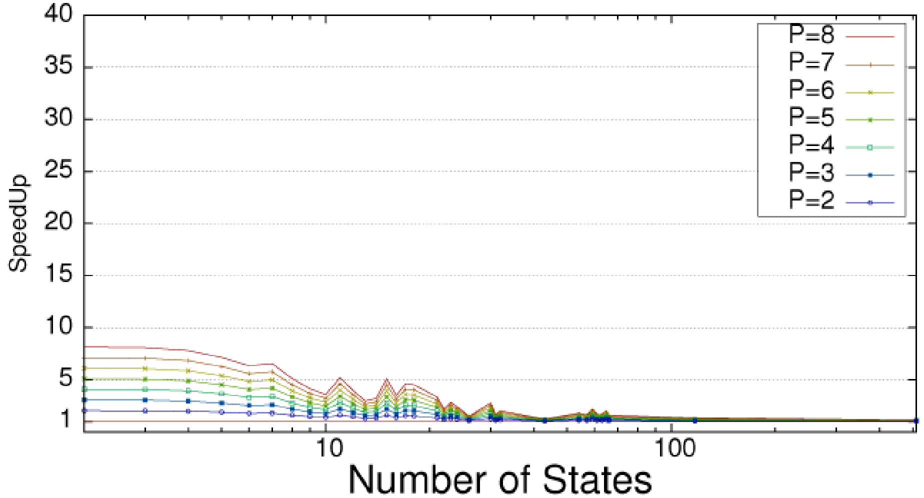


(a) Speedup over ScanProsite

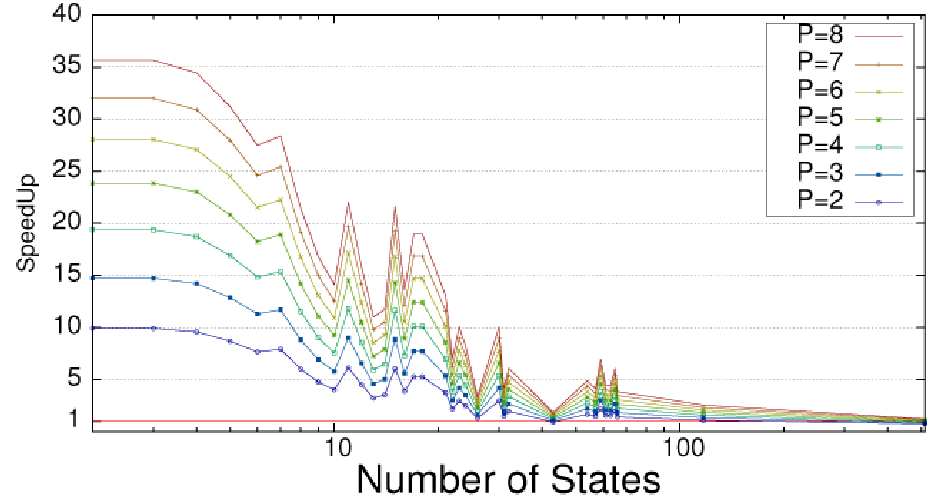


(b) Speedup over the UNIX grep utility

SIMD

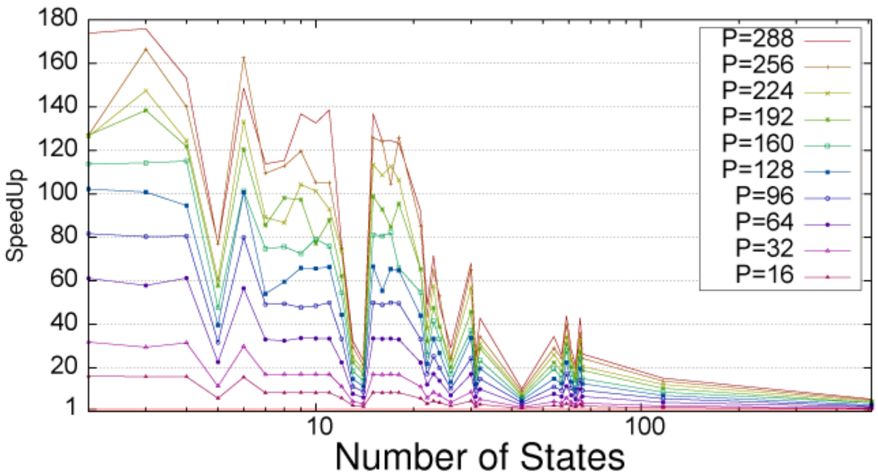


(a) Speedup without AVX2 on PCRE

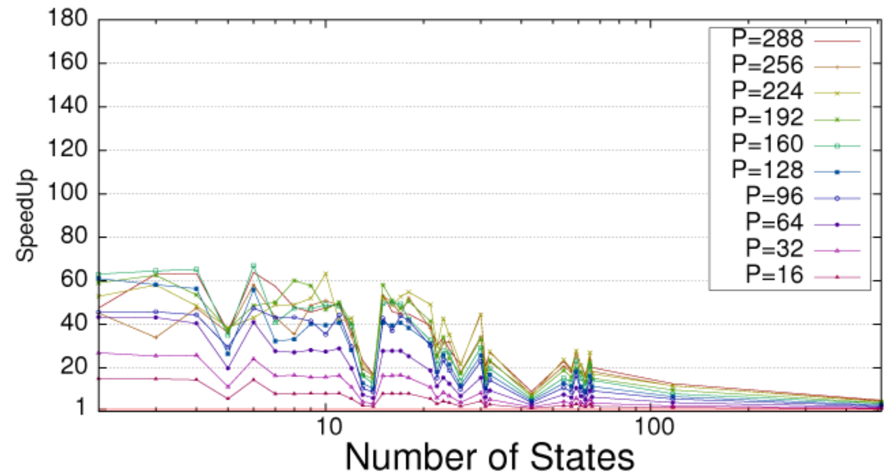


(b) Speedup with AVX2 on PCRE

Cloud Computing Environment



(a) Speedup without MPI communication cost on PCRE



(b) Speedup with MPI communication cost on PCRE

Thanks!