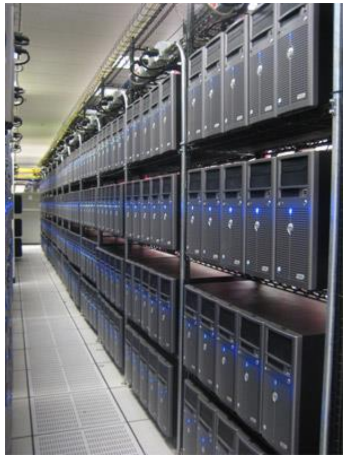
Microsoft Research Faculty Summit

Parallelizing Sequential Algorithms

Madan Musuvathi
Microsoft Research

joint work with
Saeed Maleki, Univ. of Illinios, Urbana-Champaign
Todd Mytkowicz, Microsoft Research

Hardware is Parallel



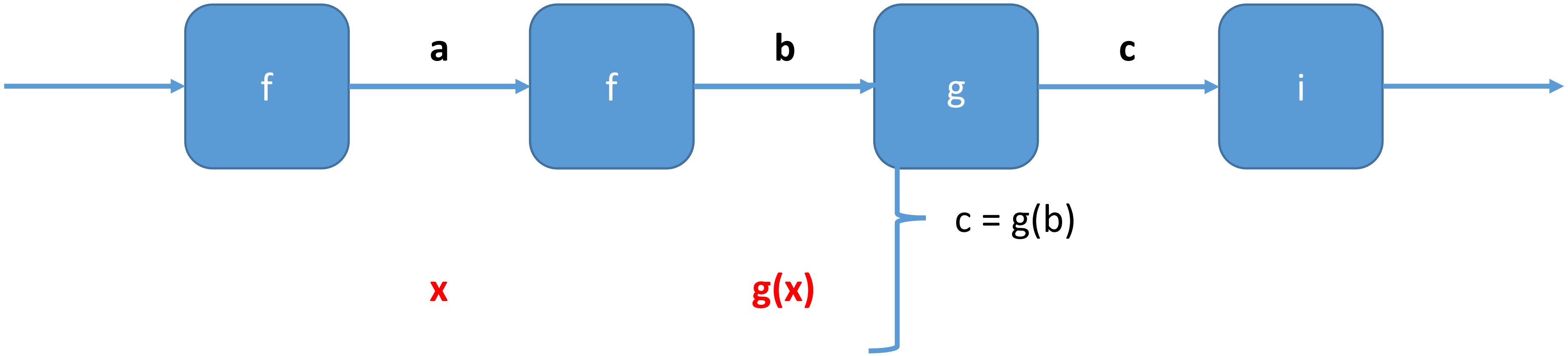
But, many important algorithms are 'inherently sequential'

Motivating Problem

Grep terabytes in seconds

- Sequentially reading terabytes from disk takes hours
- Grep implementations are sequential

Breaking Sequential Dependences



Two Parallel Algorithms

Finite State Machines

30x faster on 12 cores

Dynamic Programming

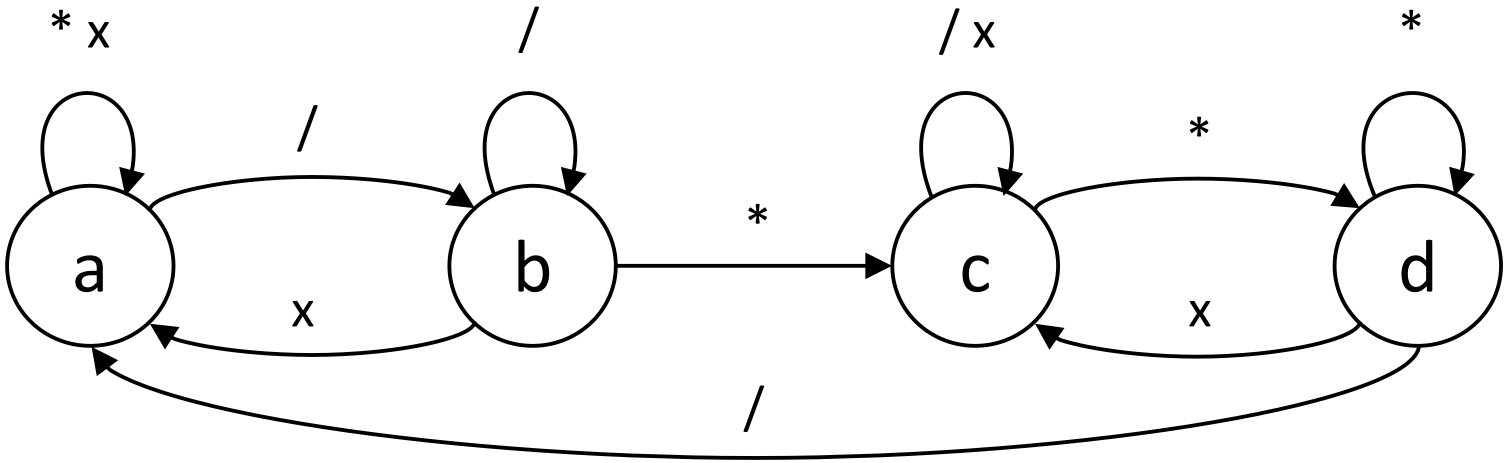
Fastest software Viterbi decoder

Higher throughput on 16 cores than a commercial FPGA implementation

Parallel Finite State Machines

FSM Applications

- grep (regex matching)
- lex (tokenization)
- Dictionary-based decoding (e.g. Huffman decoding)
- Text encoding/decoding (e.g. UTF8)

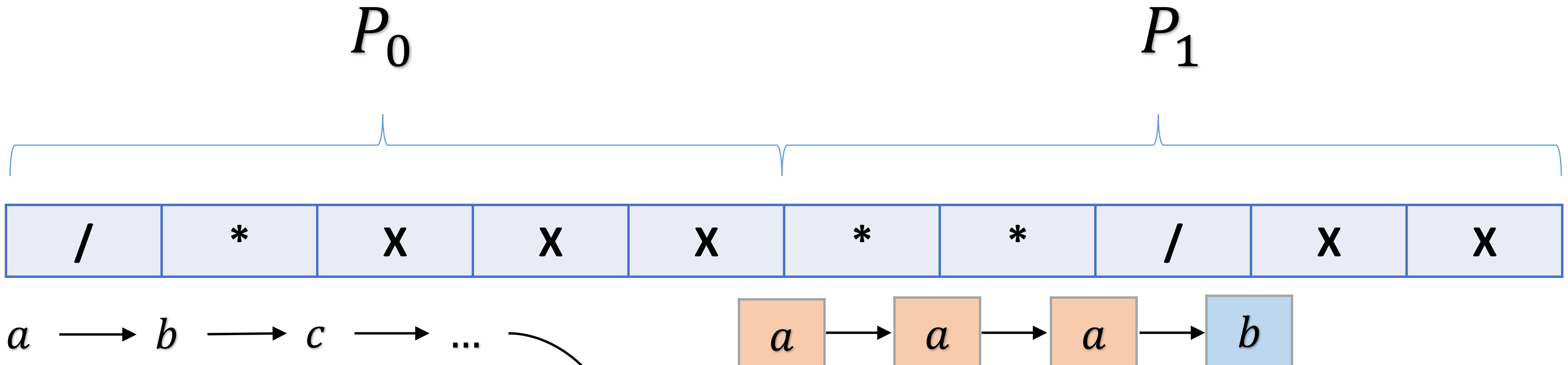


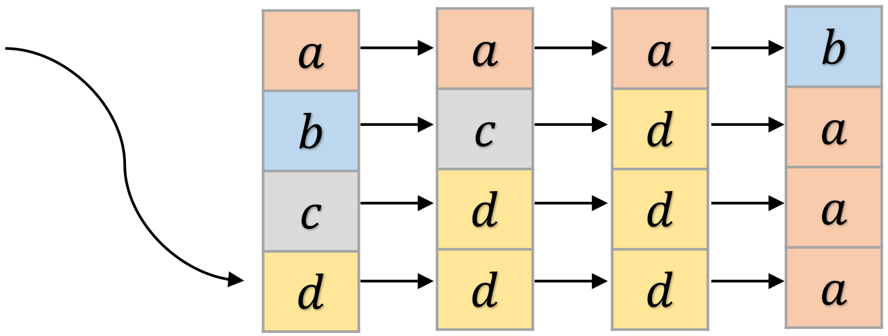
Data Dependence limits ILP, SIMD, and multicore parallelism

T	/	*	x
a	b	a	a
b	b	c	a
c	c	d	d
d	a	d	c

```
state = a;
foreach(input in)
  state = T[in][state];
```

Breaking Dependences with Enumeration

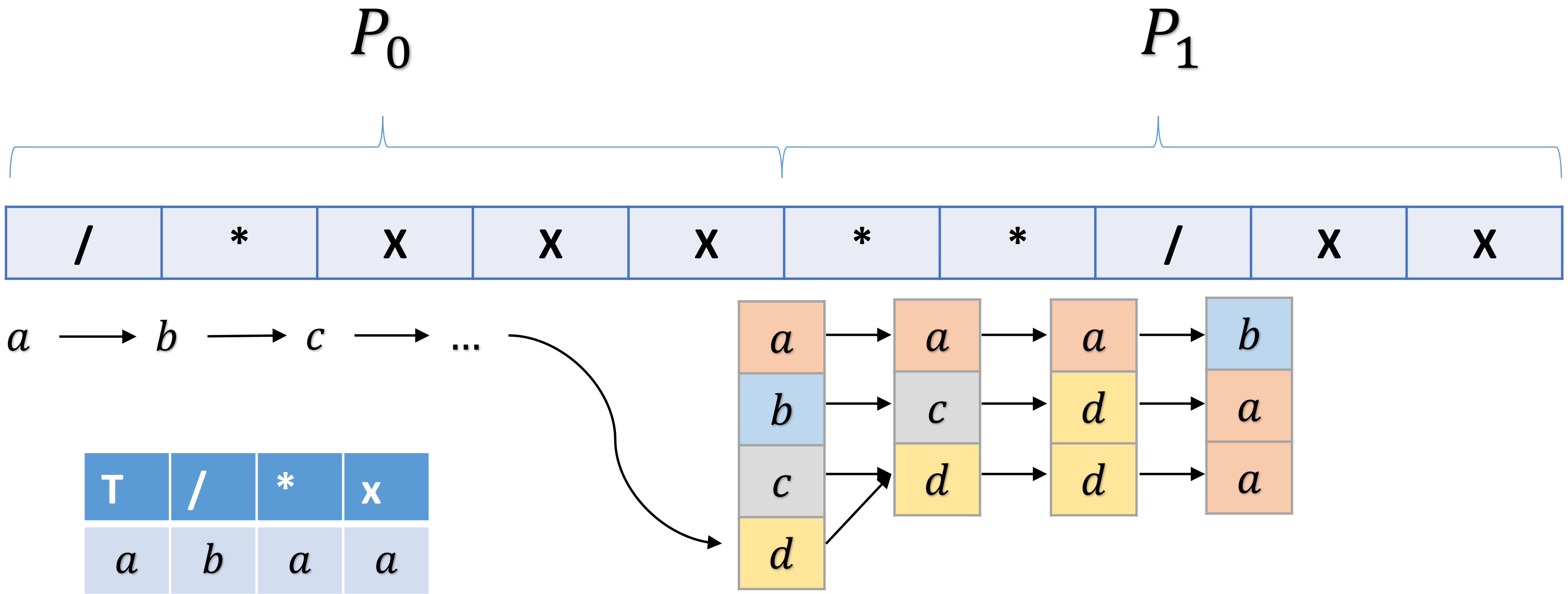




T	/	*	x
a	b	a	a
b	b	c	a
c	c	d	d
d	a	d	c

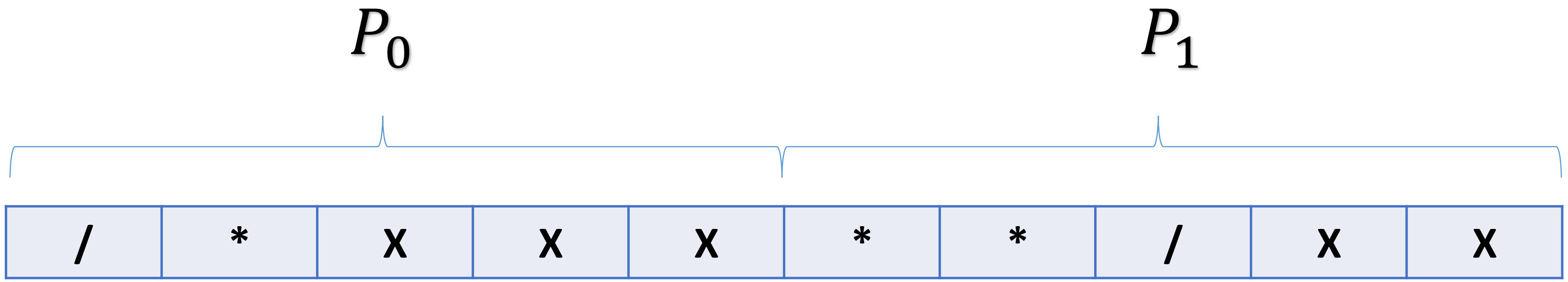
Cost of enumeration $\propto$ number of states

Convergence Reduces the Cost of Enumeration

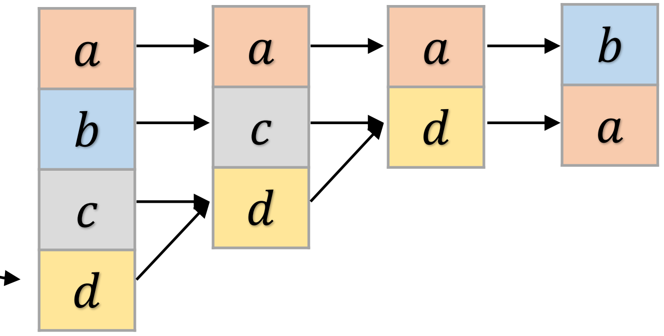


T	/	*	x
a	b	a	a
b	b	c	a
c	c	d	d
d	a	d	c

Convergence Reduces the Cost of Enumeration



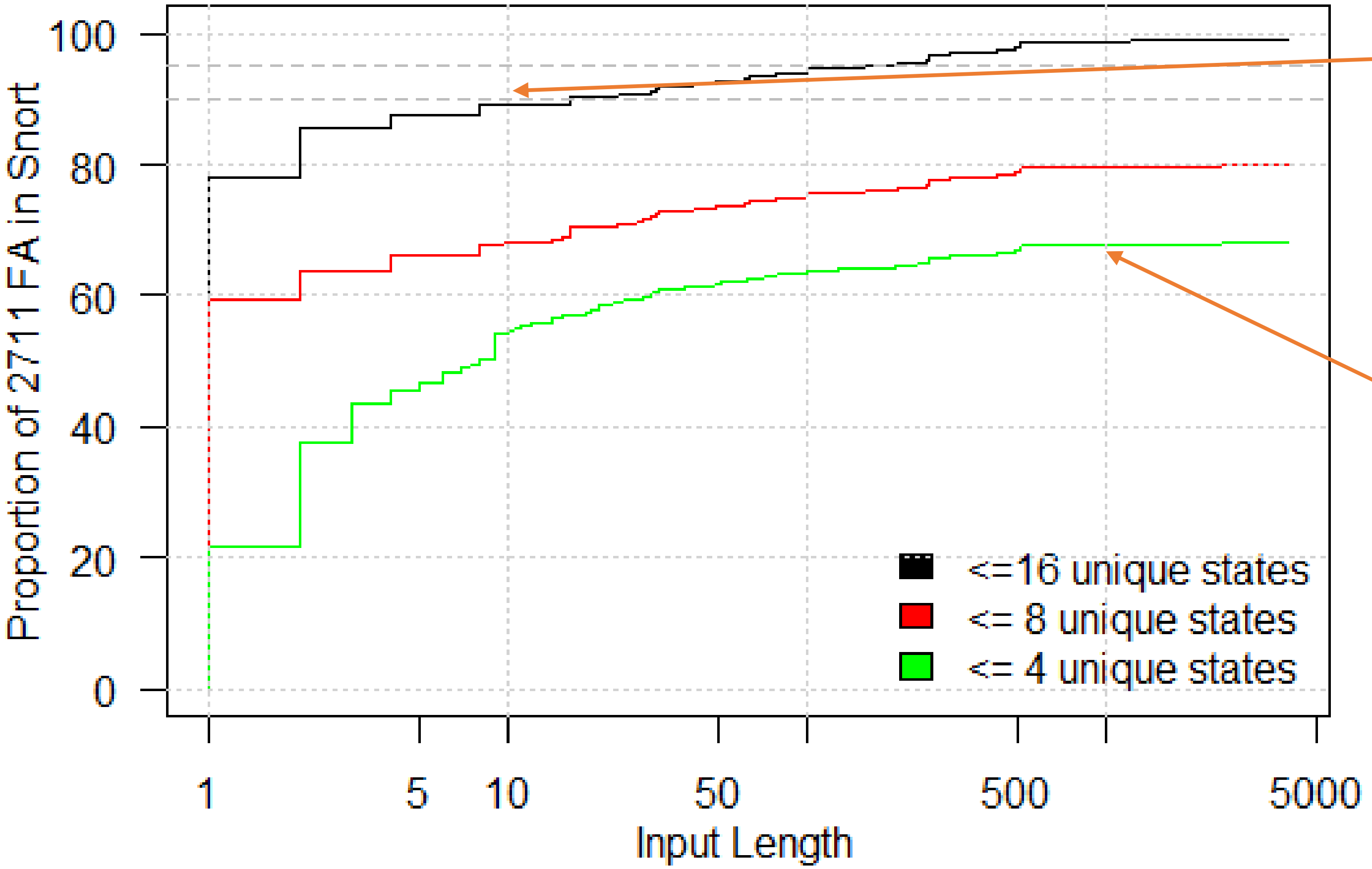
$a \longrightarrow b \longrightarrow c \longrightarrow \dots$



T	/	*	x
a	b	a	a
b	b	c	a
c	c	d	d
d	a	d	c

Cost of enumeration $\propto$ number of unique states

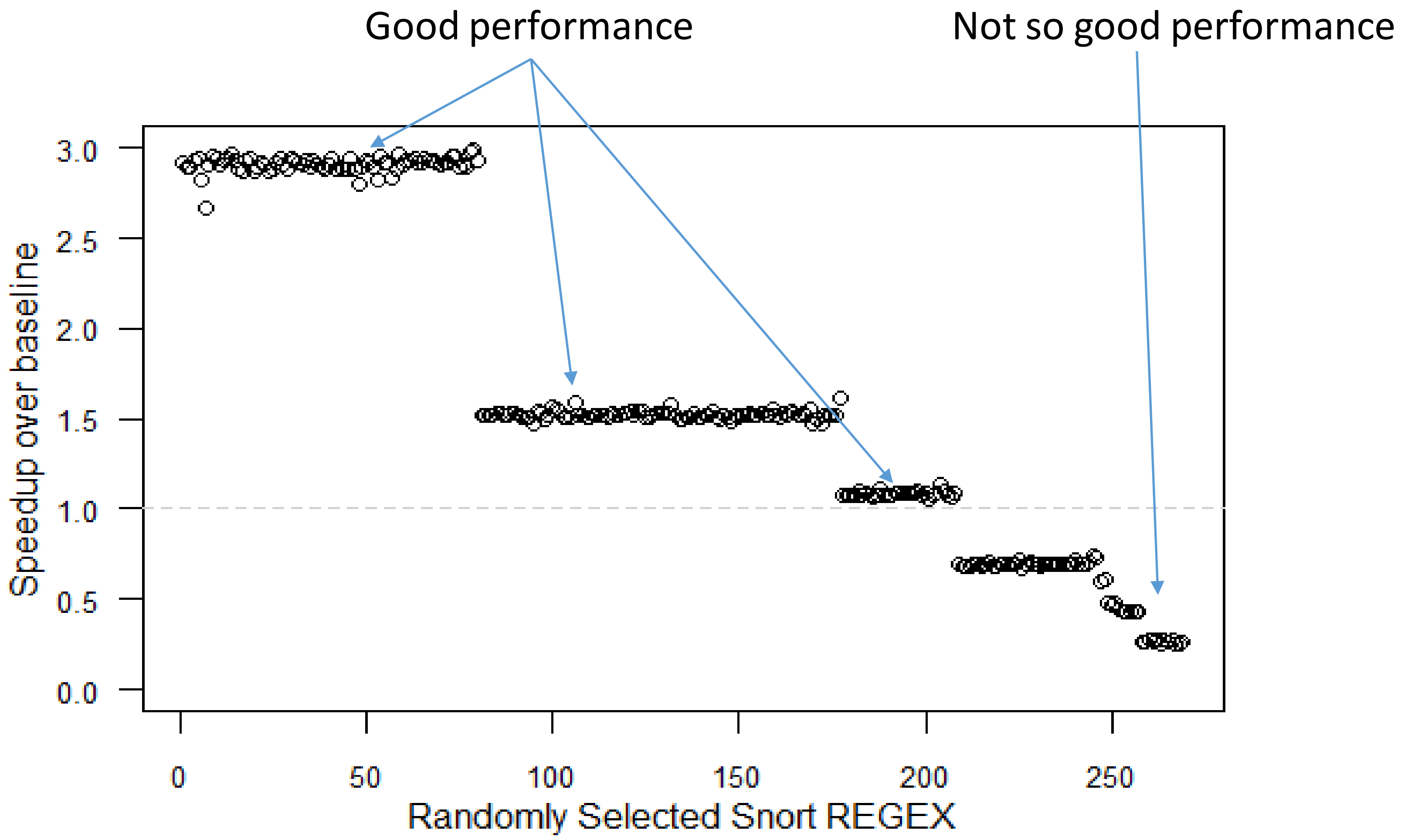
Convergence for Worst-Case Inputs



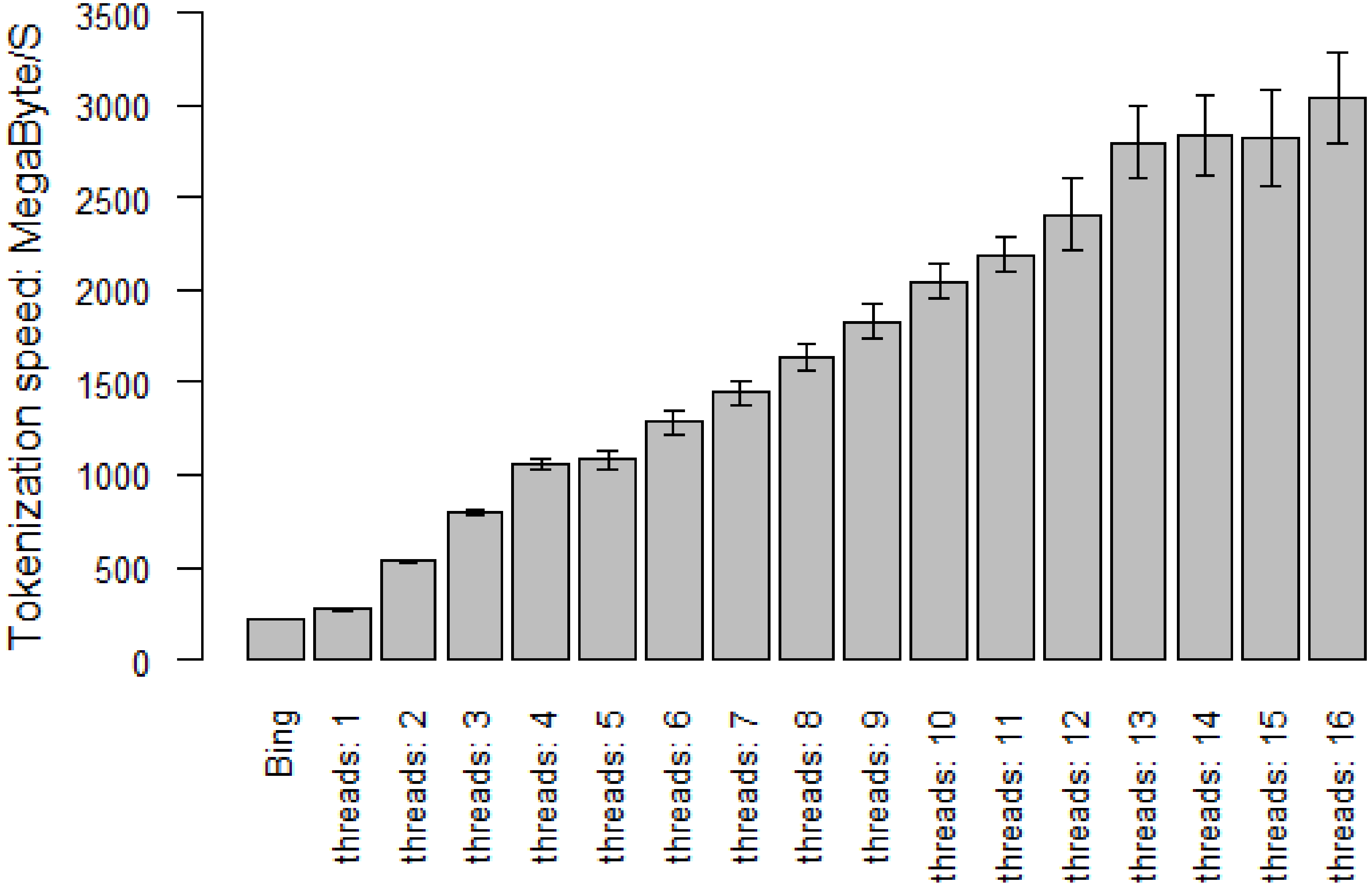
90% of FSMs converge to <= 16 states after 10 steps

Only 65% of FSMs converge to <= 4 states even after 5000 steps

Single-Core Performance when using SIMD

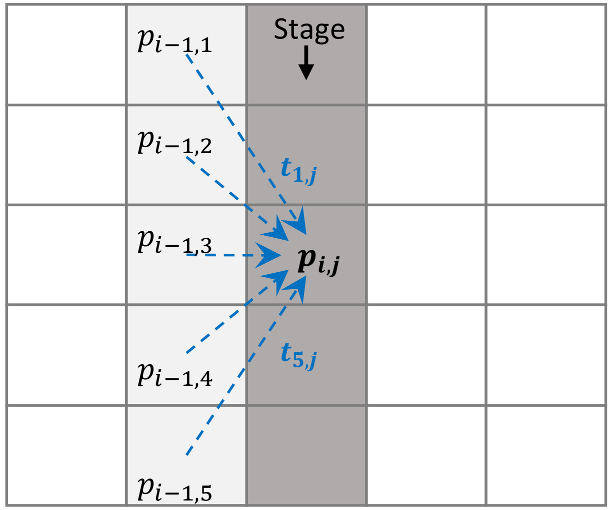


Multicore Performance for Bing Tokenization



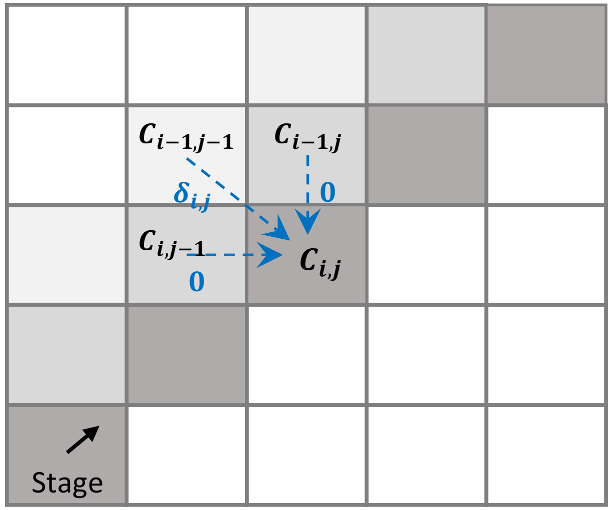
Parallelizing Dynamic Programming

Optimization Problems Solved using Dynamic Programming



$$p_{i,j} = \max_k (p_{i-1,k} * t_{k,j})$$

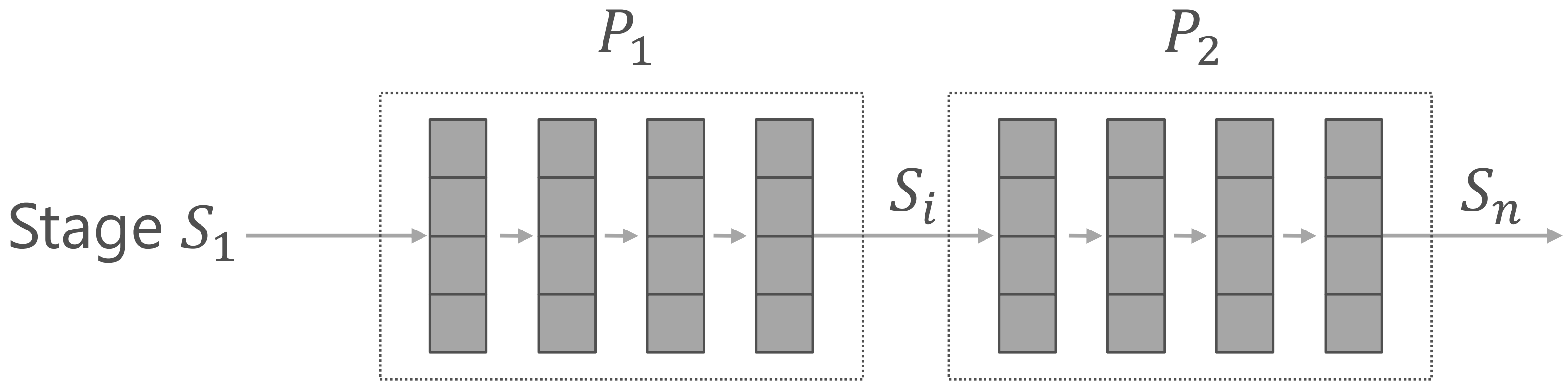
Viterbi



$$C_{i,j} = \max \begin{cases} C_{i-1,j-1} + \delta_{i,j} \\ C_{i,j-1} \\ C_{i-1,j} \end{cases}$$

diff

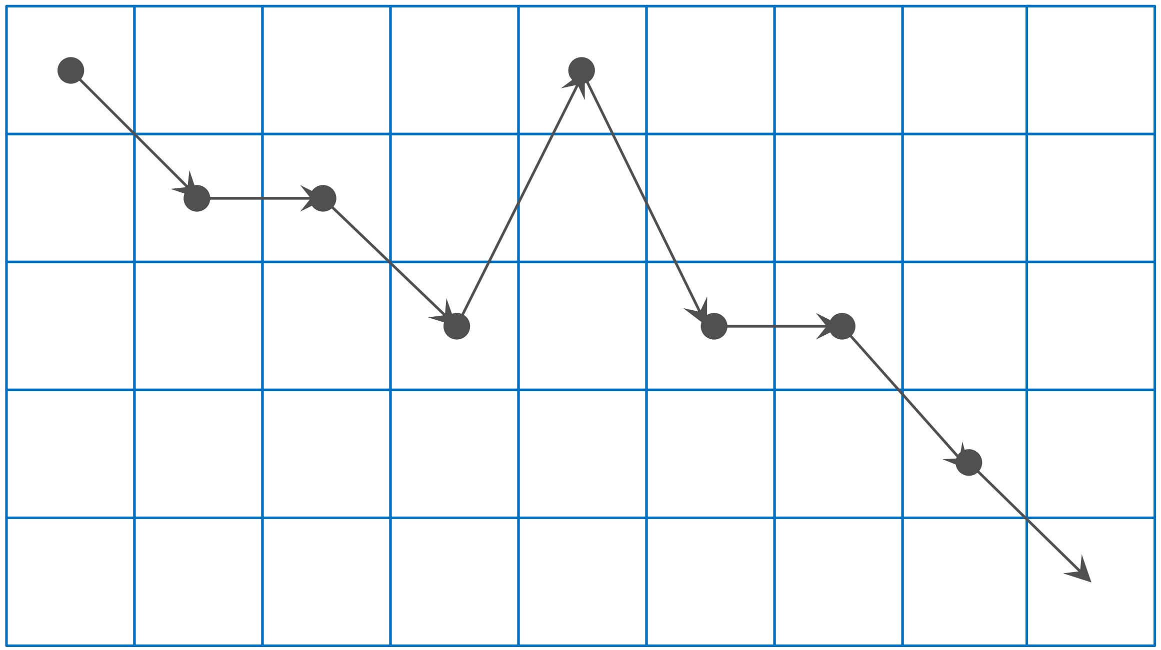
Parallelize Across Stages



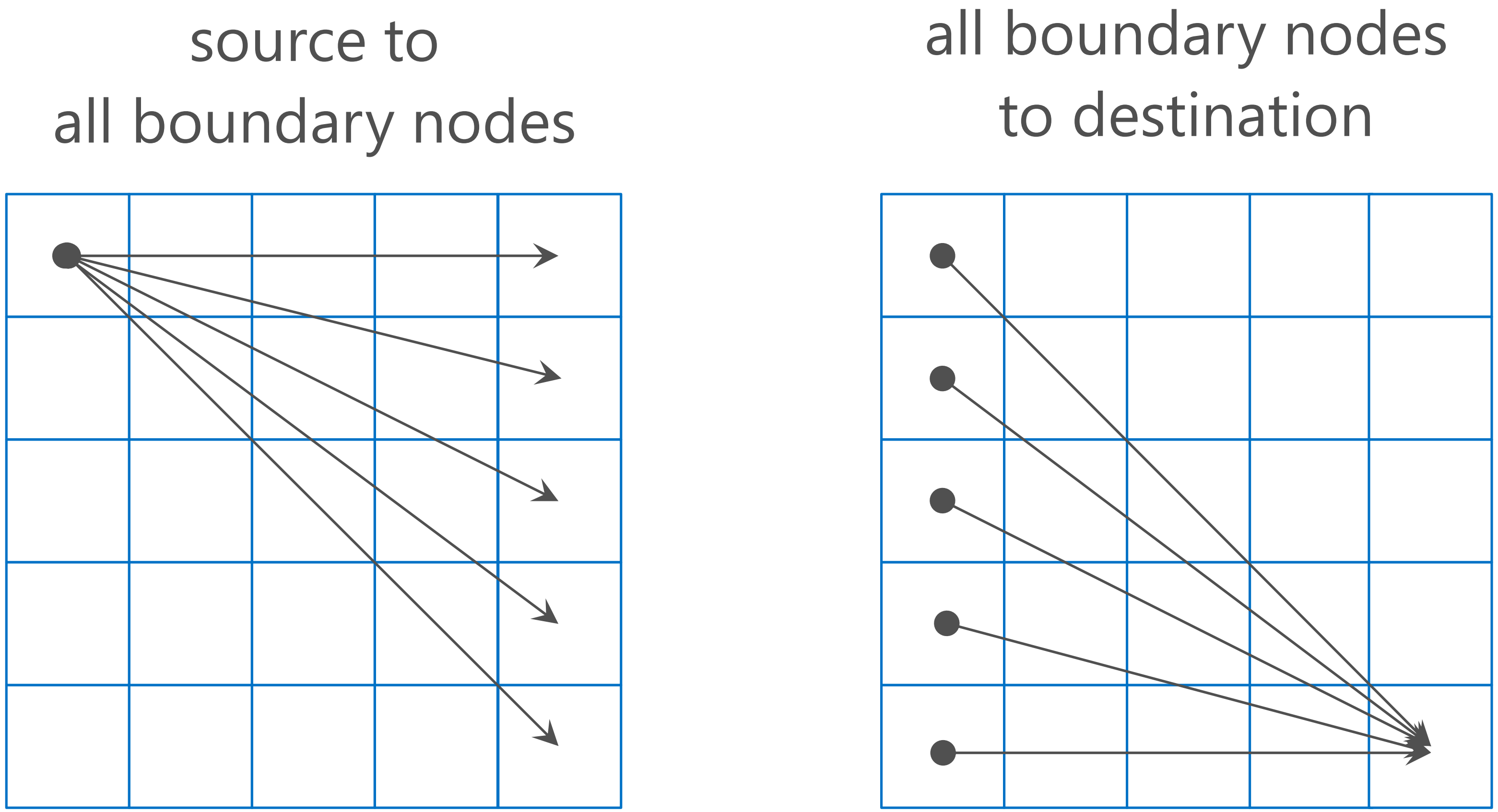
Assume recurrence is of the form

$$s_i[j] = \min_k \left(C_{ijk} + s_{i-1}[k]\right)$$

Optimization problem = Finding shortest path in some graph

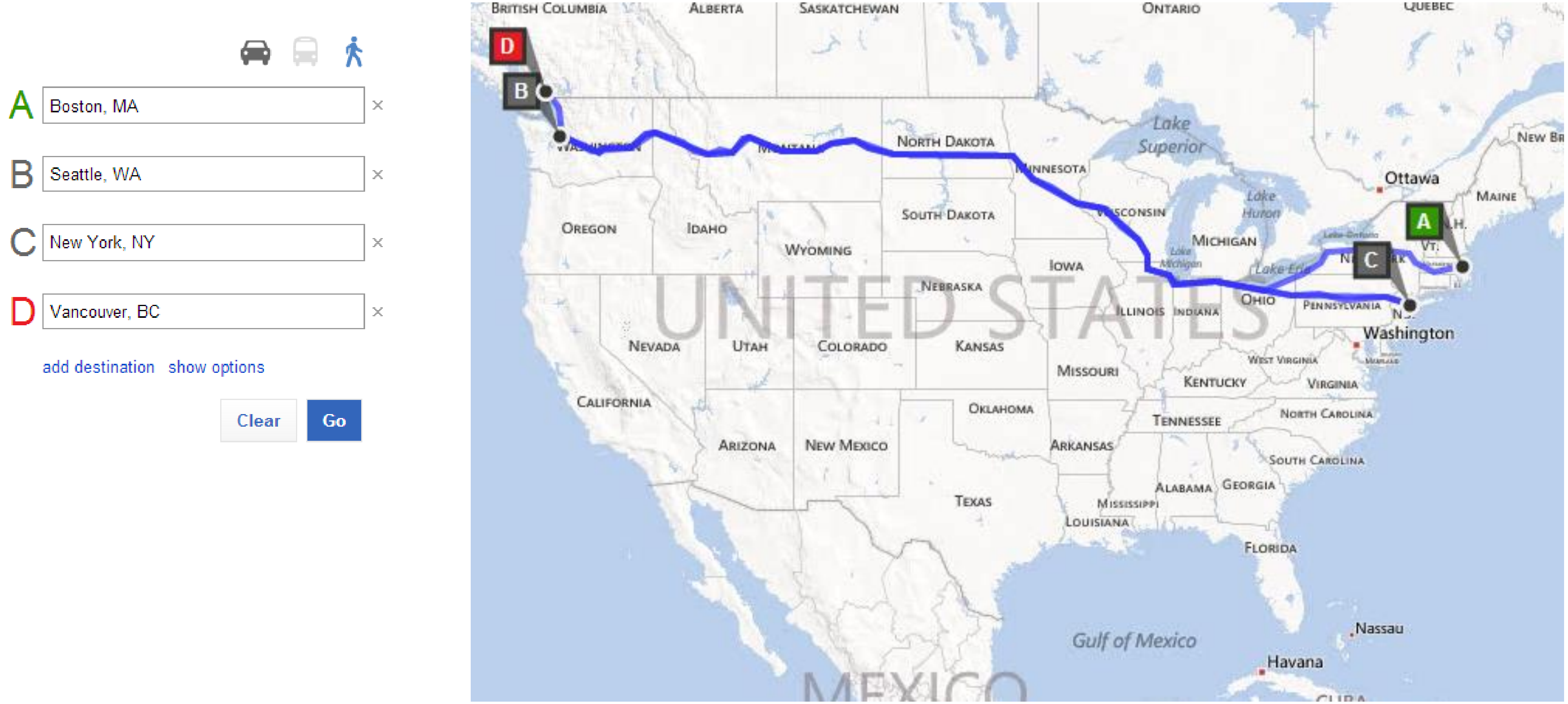


Break Dependences with All-Pair Shortest Paths

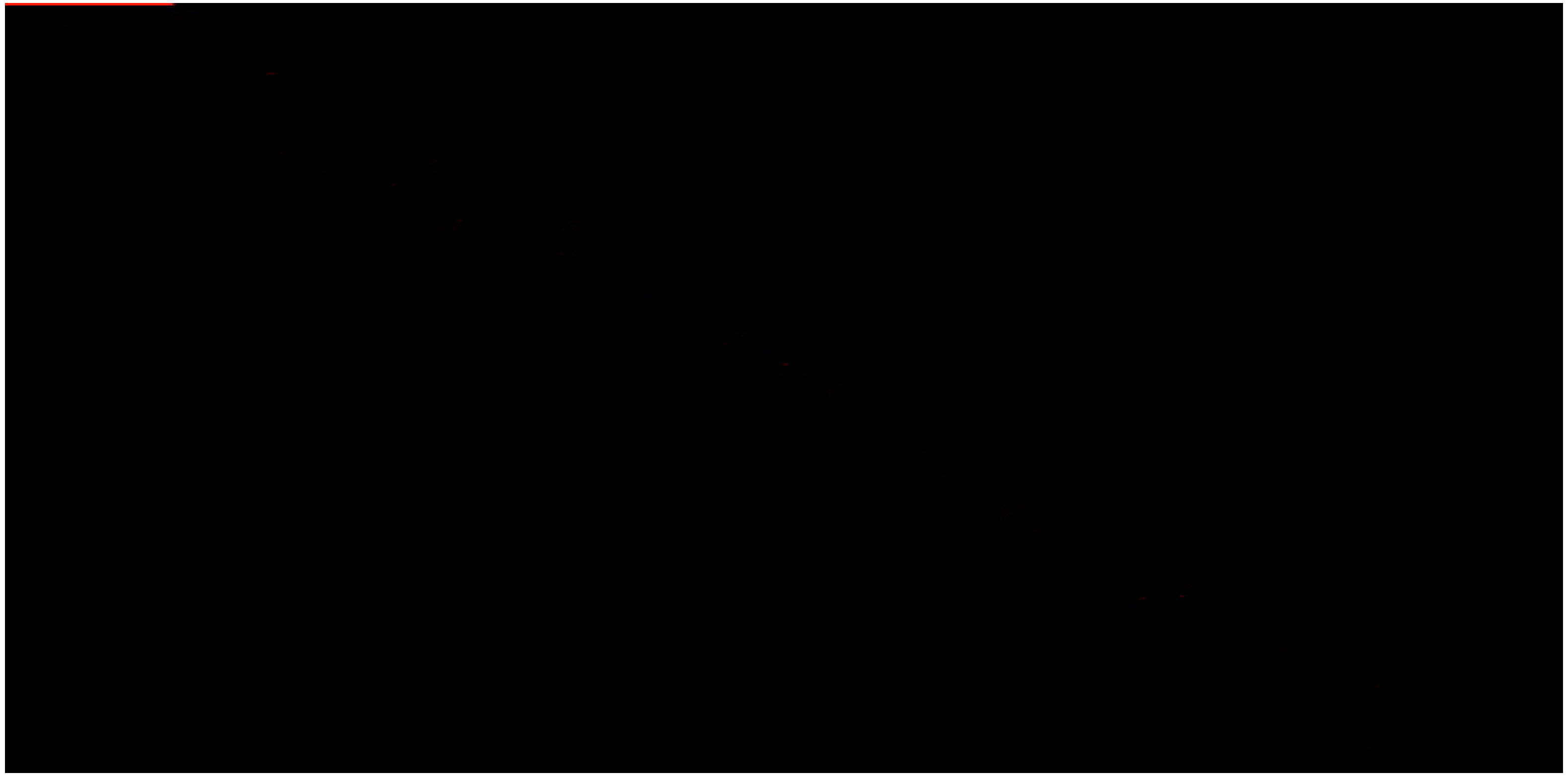


Overhead $\propto$ stage size

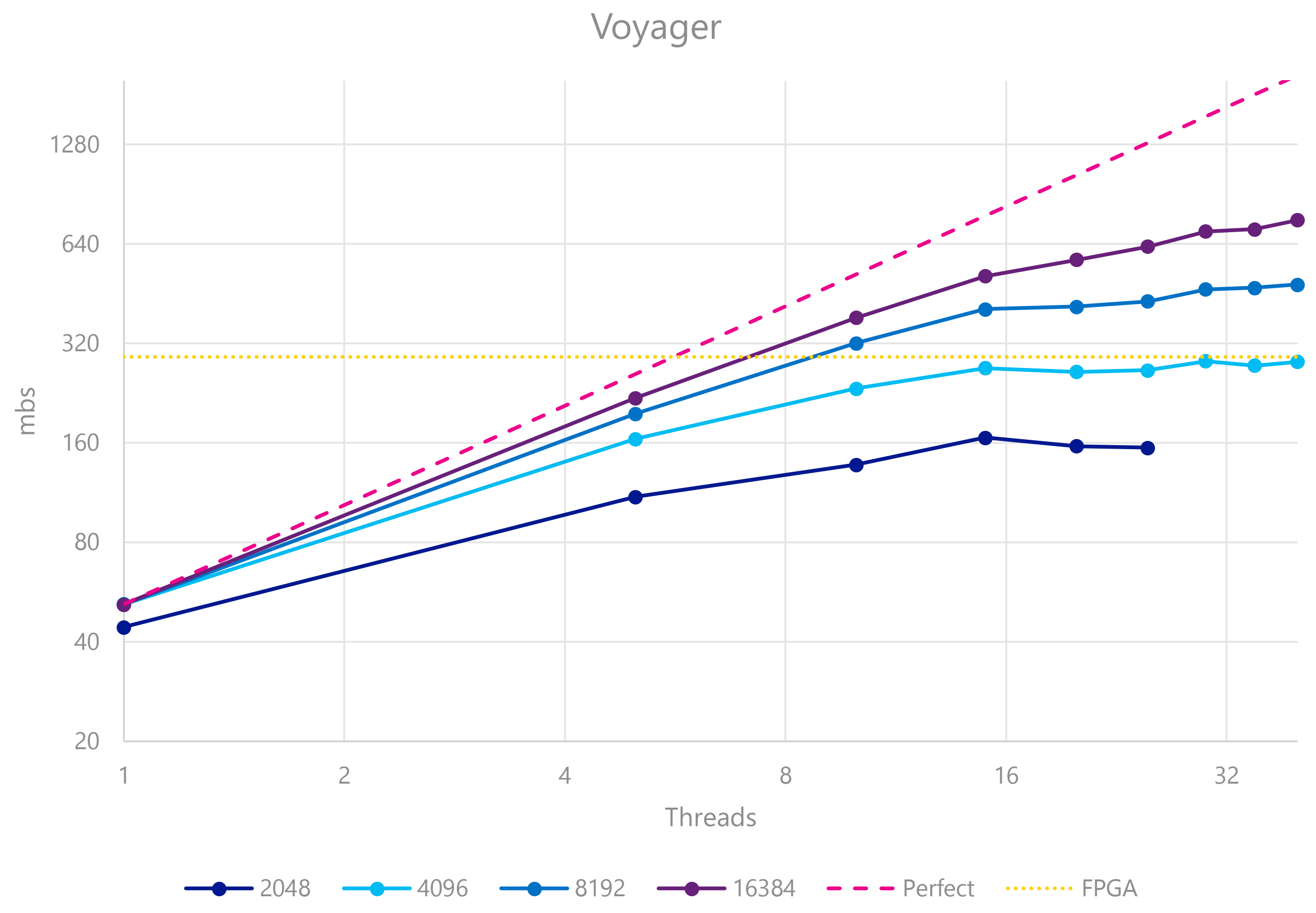
All-Pair Shortest Paths Converge



Convergence in LCS



Results – Viterbi Decoder



Conclusions

Parallel algorithms for FSMs and dynamic programming

Using dynamic convergence properties

Can we break dependences for other algorithms?

Parsing

Iterative machine learning

Graph algorithms

Can we automatically parallelize across dependences?

Save the planet and return your name badge before you leave (on Tuesday)

Microsoft Privacy Policy statement applies to all information collected. Read at research.microsoft.com

Microsoft

© 2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.