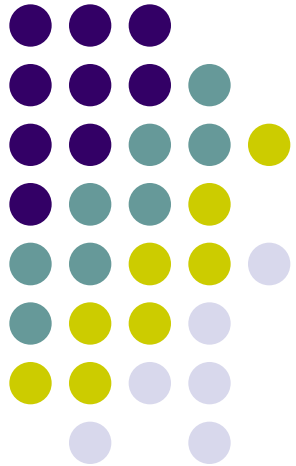
Adaptive Software Speculation
for Enhancing the Cost-Efficiency of Behavior-Oriented Parallelization

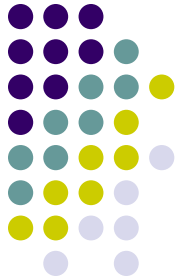


Yunlian Jiang

Xipeng Shen

The College of William and Mary

High-level Parallelism



- Parallel computing is becoming ubiquitous
- High-level parallelism exists in many programs
  - E.g. utilities, interpreters, scientific computations
  - Difficult to parallelize

Complexity in the code

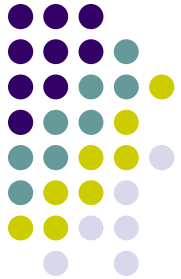
Bit-level operations,
unrestricted pointers,
exception handling,
custom mem. management,
third-party libraries

Uncertain parallelism

Example*:

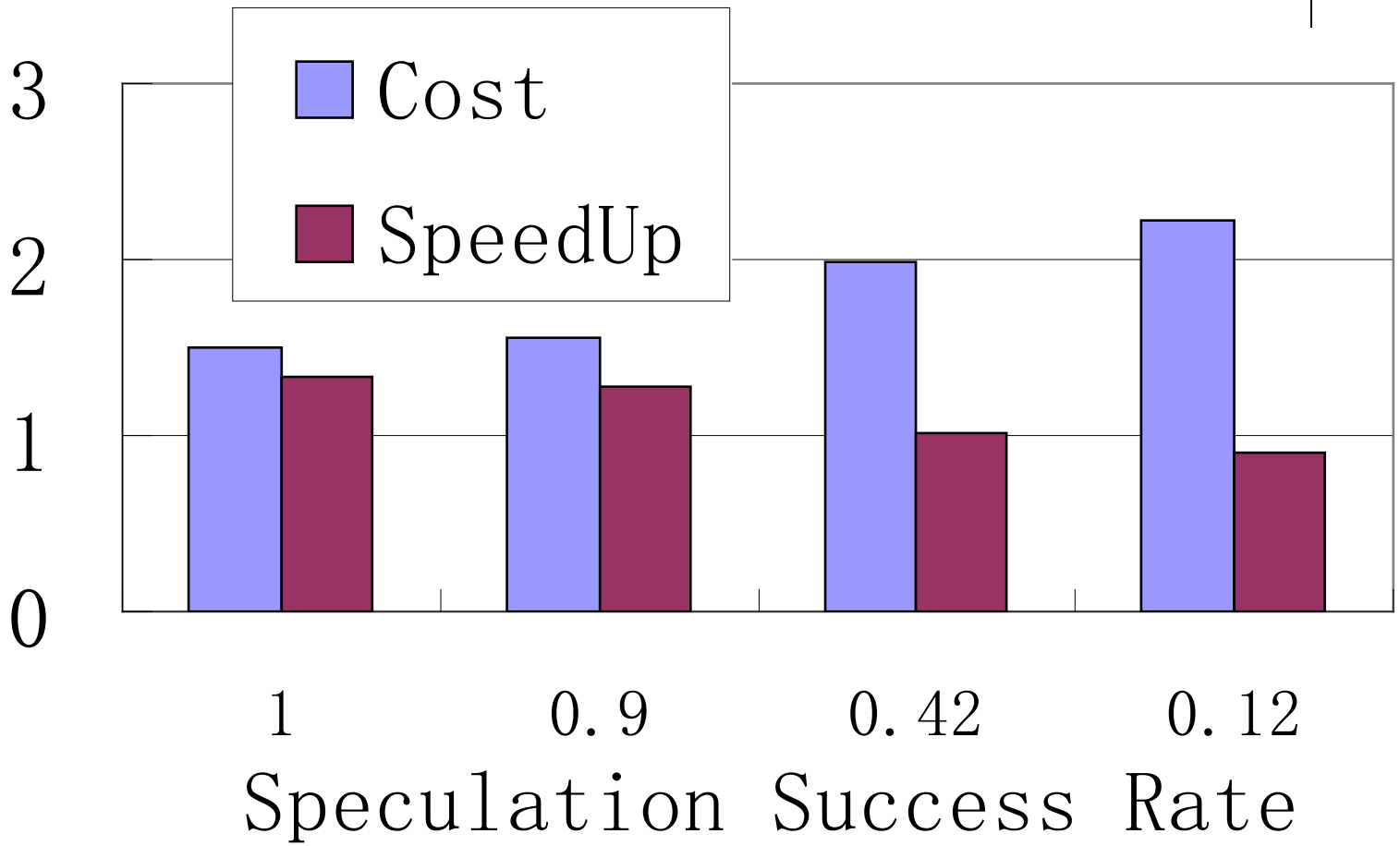
```
 while ( s=nextSentence() )
{  parse(s);
 if ( isCommand(s) )
        updateParsingEnv(s);
}
```

Software Behavior-Oriented Parallelization [Ding+:PLDI07]

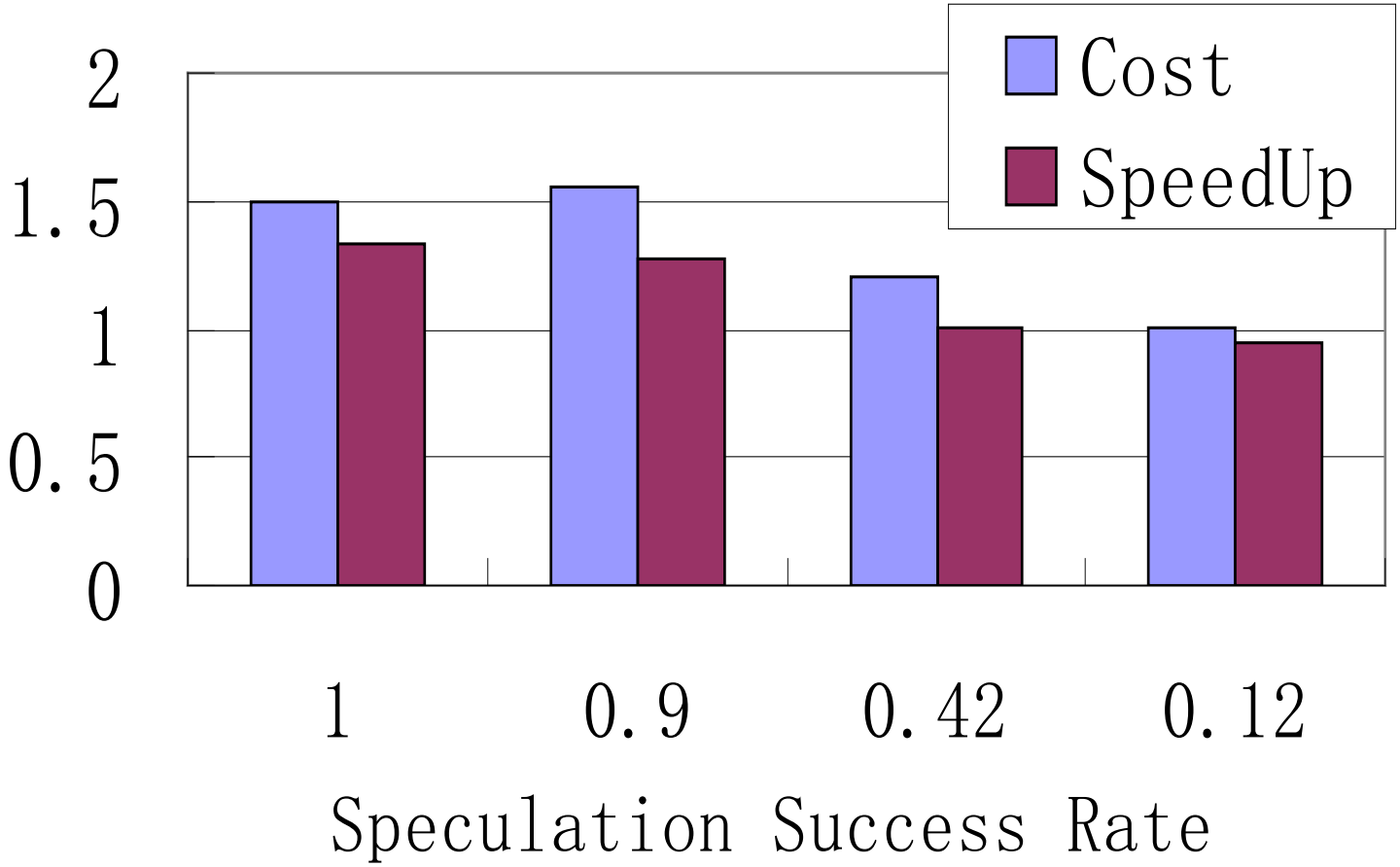


- Speculatively execute programs in parallel
- Efficiently detect dependence during runtime
- But, blind speculation causes cost-inefficiency.

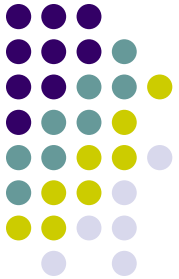
Cost and Speedup



Cost and Speedup

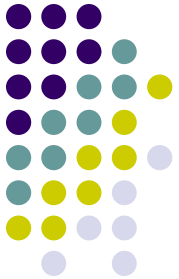


Outline



- Introduction to BOP
- Adaptive BOP
- Experimental Result
- Conclusion

Outline



- Introduction to BOP
- Adaptive BOP
- Experimental Result
- Conclusion

Behavior-Oriented Parallelization (BOP)

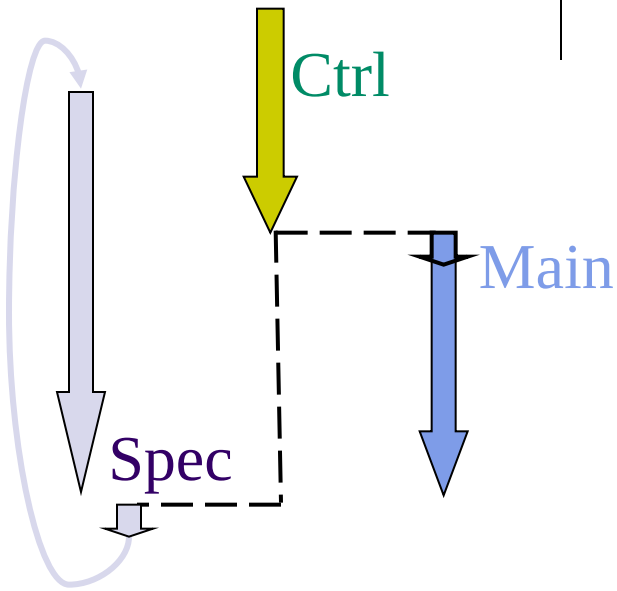
- A tool for parallelizing sequential programs
- Need no parallel programming or debugging
- Basic scheme: software speculation
- Correctness protected through runtime system

Basic Scheme of BOP

Parser

```
. . . . . .
While (S= ReadSentence())
{
    BeginPPR();
    Parse(S);
    EndPPR();
}
. . . . . .
```

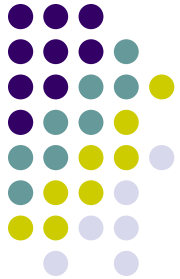
Possibly Parallel Region (PPR)



Two reason for failed speculation:

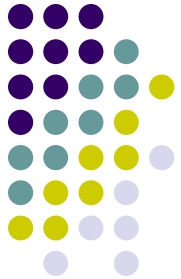
1. Dependence violation
2. Spec runs too slow

Problem to tackle



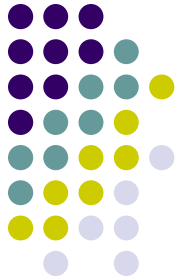
- Cost inefficiency
  - BOP blindly speculates every PPR instance
  - Failed speculation may
    - Cause slowdown to applications
      - Protection overhead
      - Resource (cache, bus) contention
    - Waste computing resources
      - CPU --- multi-programming environment
      - Power --- Mobile computing

Outline



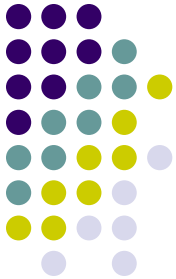
- Introduction to BOP
- Adaptive BOP
- Experimental Result
- Conclusion

Solution: Adaptive Speculation



- Basic strategy
  - Predict profitability of PPR
  - Speculate only likely profitable ones
- Prediction approaches
  - Extended last-value-based
  - Decayed-history-based

Extended last-value-based prediction



- Speculate or not?
  - Speculate only if PPRsToSkip == 0.
- Adjust PPRsToSkip
  - If this PPR is not speculated
    - PPRsToSkip - -
  - On a failed speculation
    - PPRsToSkip = NextPenalty;
    - NextPenalty *= **α;** *(increase penalty exponentially)*
  - On a successful speculation
    - NextPenalty = 1; *(reset the penalty on the next failure)*

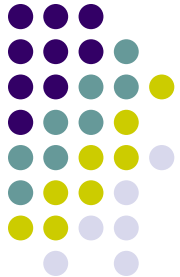
Extended last-value-based prediction

| : Profitable PPR | : Unprofitable PPR

Speculation!
Success
NextPenalty = 1

- PPRsToSkip=0
- NextPenalty=1
- α=2

Extended last-value-based prediction



| : Profitable PPR | : Unprofitable PPR

Speculation!
Failed
PPRsToSkip = 1
NextPenalty = 2

- PPRsToSkip=0
- NextPenalty=1
- α=2

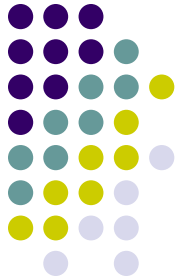
Extended last-value-based prediction

| : Profitable PPR | : Unprofitable PPR

No Speculation!
PPRsToSkip = 0

- PPRsToSkip=1
- NextPenalty=2
- α=2

Extended last-value-based prediction



| : Profitable PPR | : Unprofitable PPR

Speculation!
Failed
PPRsToSkip = 2
NextPenalty = 4

- PPRsToSkip=0
- NextPenalty=2
- α=2

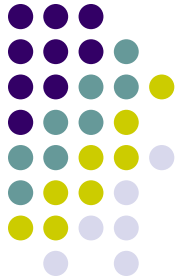
Extended last-value-based prediction

| : Profitable PPR | : Unprofitable PPR

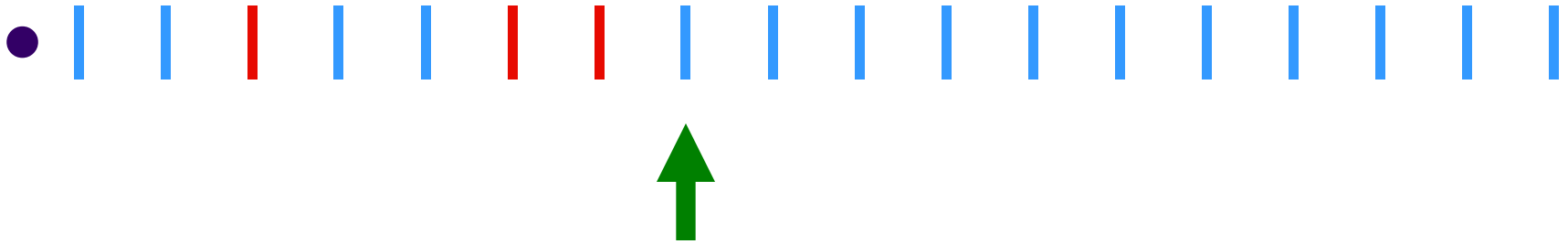
No Speculation!
PPRsToSkip = 1

- PPRsToSkip=2
- NextPenalty=4
- α=2

Extended last-value-based prediction



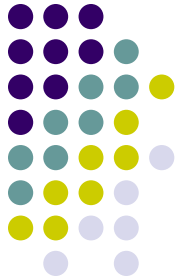
| : Profitable PPR | : Unprofitable PPR



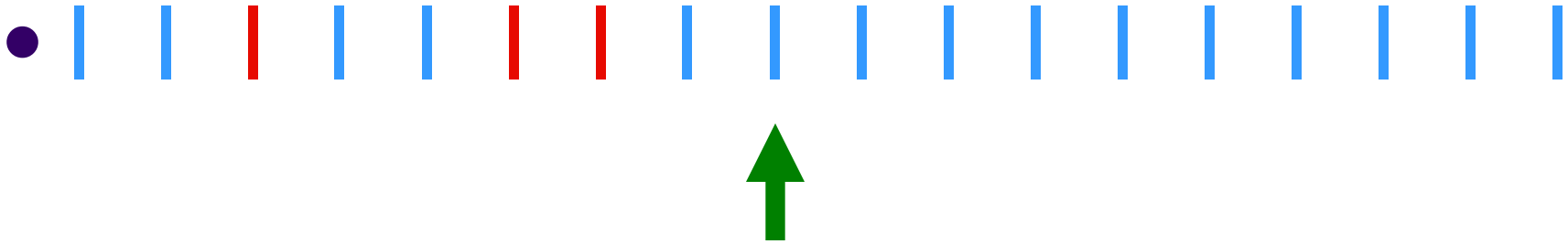
No Speculation!
PPRsToSkip = 0

- PPRsToSkip=1
- NextPenalty=4
- α=2

Extended last-value-based prediction



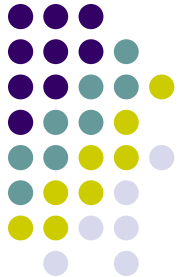
| : Profitable PPR | : Unprofitable PPR



Speculation!
Success
NextPenalty = 1

- PPRsToSkip=0
- NextPenalty=4
- α=2

Extended last-value-based prediction



| : Profitable PPR | : Unprofitable PPR

Speculation!
Success
NextPenalty = 1

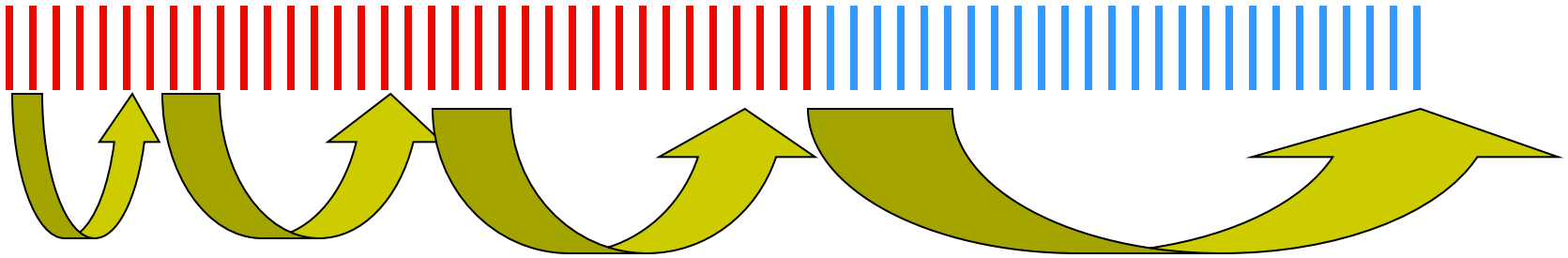
- PPRsToSkip=0
- NextPenalty=1
- α=2

Extended last-value-based prediction

- Limitations
  - Can not keep history well
    - Successful speculation

      →clean history

  - Phase changes



Decayed-history-based prediction

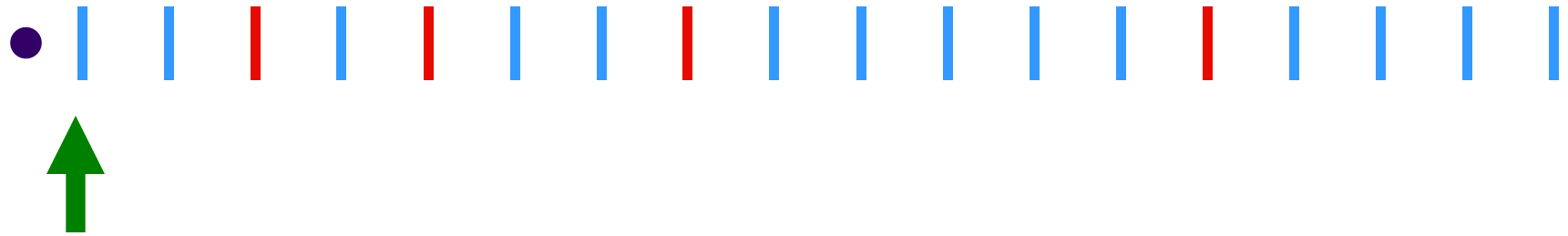
- Cumulative gain (*CG*)
  - $cg = \gamma * g + (1-\gamma) * cg$
- Expected Profitability (*EP*)
  - $EP = cg + SkippedPPRs * \beta$
  - Speculate only if $EP > TH_{EP}$

$g = \begin{cases} 1 : \text{success} \\ 0 : \text{failed} \end{cases}$

* TH_{EP} : threshold of speculation;
* SkippedPPRs: reset to 0 on a success
  increase by 1 on a non-speculated PPR

Decayed-history-based prediction

| : Profitable PPR | : Unprofitable PPR

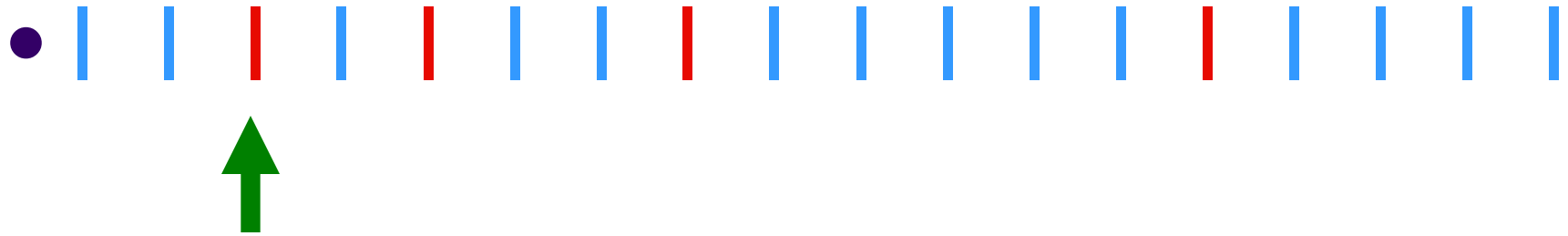


- TH_{EP}=0.6
- β=0.2
- γ = 0.5
- SkippedPPRs=0
- cg=1
- EP =1

Speculation!
Success
cg = 1
EP = 1

Decayed-history-based prediction

| : Profitable PPR | : Unprofitable PPR

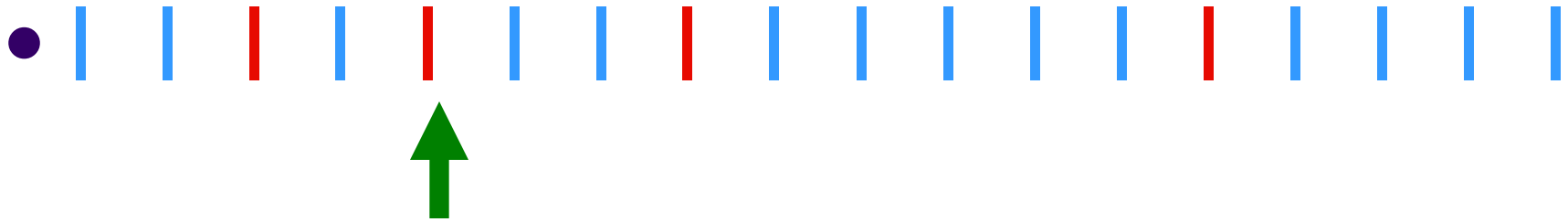


- TH_{EP}=0.6
- β=0.2
- γ = 0.5
- SkippedPPRs=0
- cg=1
- EP =1

Speculation!
Failed
cg = 0.5
EP = 0.5

Decayed-history-based prediction

| : Profitable PPR | : Unprofitable PPR

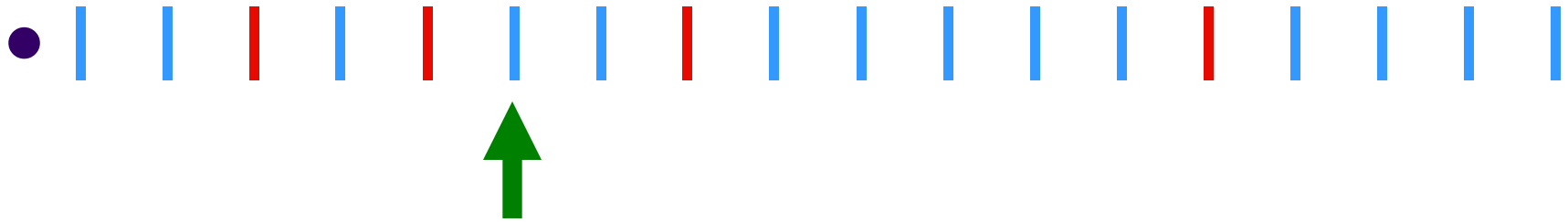


- TH_{EP}=0.6
- β=0.2
- γ = 0.5
- SkippedPPRs=0
- cg=0.5
- EP =0.5

No Speculation!
SkippedPPRs=1
EP = 0.7

Decayed-history-based prediction

| : Profitable PPR | : Unprofitable PPR

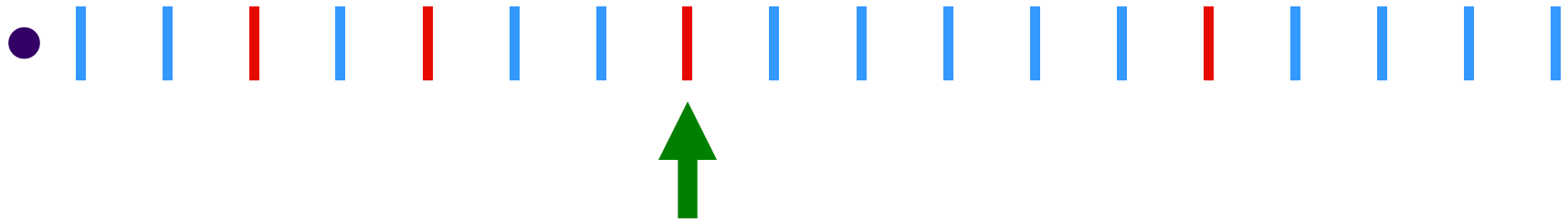


- TH_{EP}=0.6
- β=0.2
- γ = 0.5
- SkippedPPRs=1
- cg=0.6
- EP =0.7

Speculation!
Success
SkippedPPRs=0
cg = 0.8
EP = 0.8

Decayed-history-based prediction

| : Profitable PPR | : Unprofitable PPR

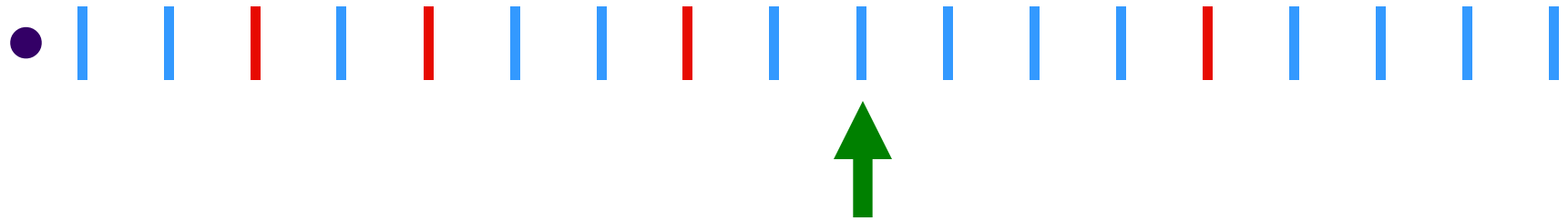


- TH_{EP}=0.6
- β=0.2
- γ = 0.5
- SkippedPPRs=0
- cg=0.8
- EP =0.8

Speculation!
Failed
cg = 0.4
EP = 0.4

Decayed-history-based prediction

| : Profitable PPR | : Unprofitable PPR

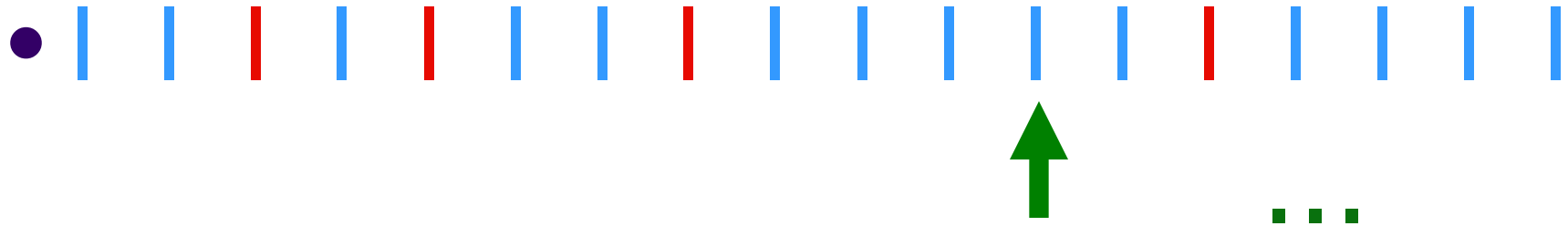


- TH_{EP}=0.6
- β=0.2
- γ = 0.5
- SkippedPPRs=0
- cg=0.4
- EP =0.4

No Speculation!
SkippedPPRs = 1
EP = 0.6

Decayed-history-based prediction

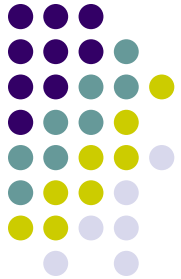
| : Profitable PPR | : Unprofitable PPR



- TH_{EP}=0.6
- β=0.2
- γ = 0.5
- SkippedPPRs=1
- cg=0.4
- EP =0.6

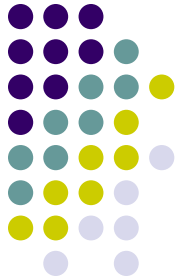
Speculation!
Success
SkippedPPRs = 0
EP = 0.7

Outline



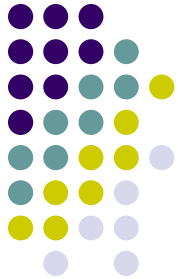
- Introduction
- Adaptive-Algorithms
- Experimental Result
- Conclusion

Experimental Result



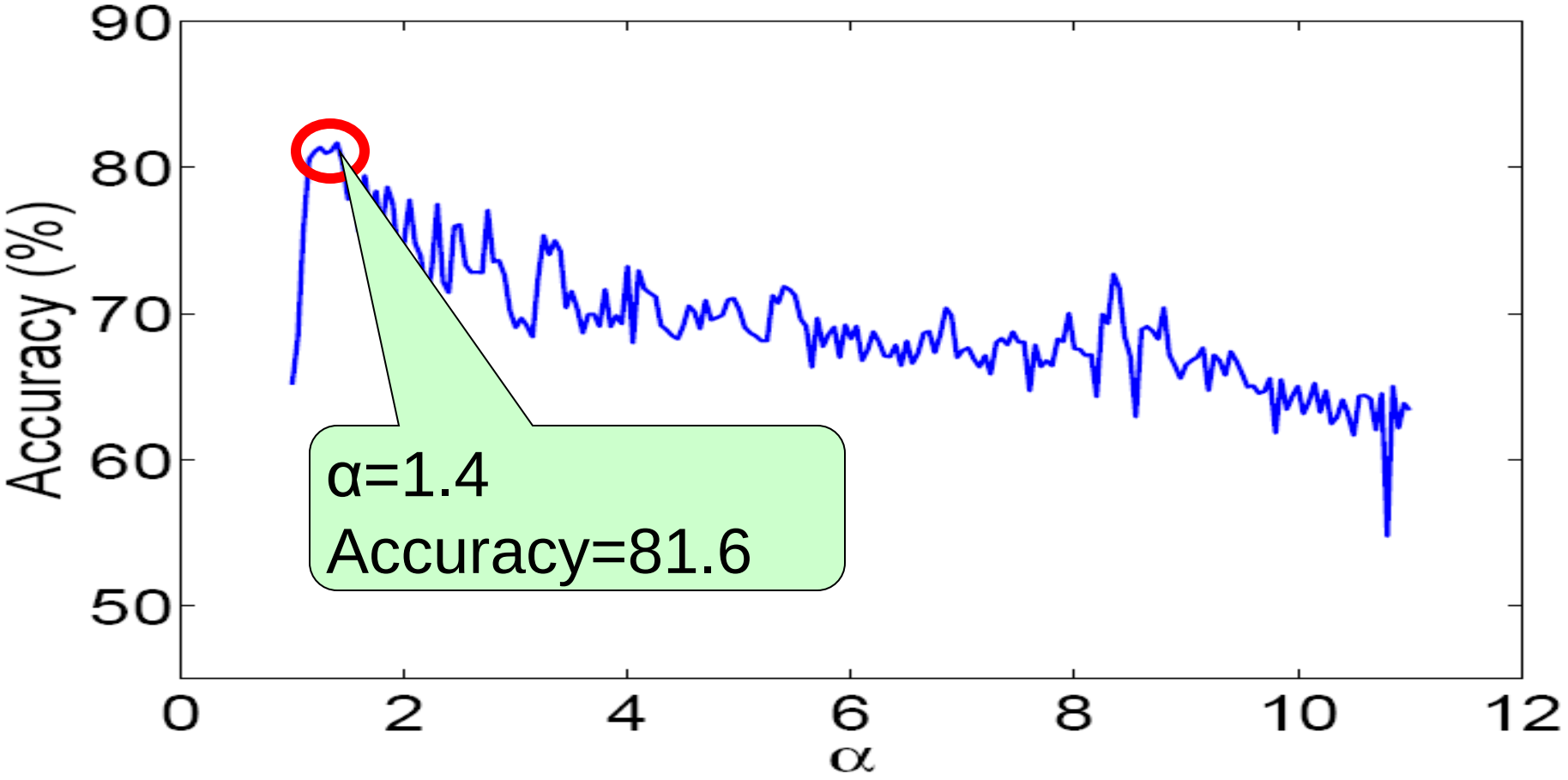
- Prediction Accuracy
  - Choose the best parameters for the algorithms
  - Evaluate two algorithms
- Computation Efficiency
  - Finishing time
  - Time spent on all CPUs (Cost)

Accuracy for Last-value-based

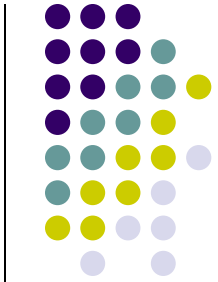


- Adjust non-speculate numbers
  - Non-speculation
    - PPRsToSkip -1
  - Success Speculation
    - NextPenalty= 1
  - Failed Speculation
    - PPRsToSkip = NextPenalty
    - NextPenalty *= **α**

Accuracy for Last-value-based



Accuracy for Decayed history based



- G_TH : gain threshold
- Current state weight
  - gain+ quota*β
  - Non-speculative execution
    - quota++
  - Speculative execution
    - gain = γ*g + (1-γ) * gain
    - quota → 0

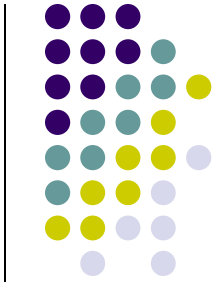
Accuracy for Decayed history based

- Cumulative gain (*CG*)
  - $cg = \gamma * g + (1-\gamma) * cg$
- Expected Profitability (*EP*)
  - $EP = cg + SkippedPPRs * \beta$
  - Speculate only if $EP > TH_{EP}$

$$g = \begin{cases} 1 : \text{success} \\ 0 : \text{failed} \end{cases}$$

* TH_{EP} : threshold of speculation;
* SkippedPPRs: reset to 0 on a success
  increase by 1 on a non-speculated PPR

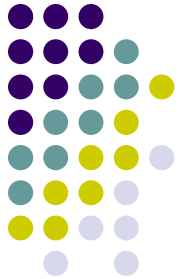
Accuracy for Decayed history based



- TH_{EP} = 0.25
- β= 0.0075
- γ = 0.4

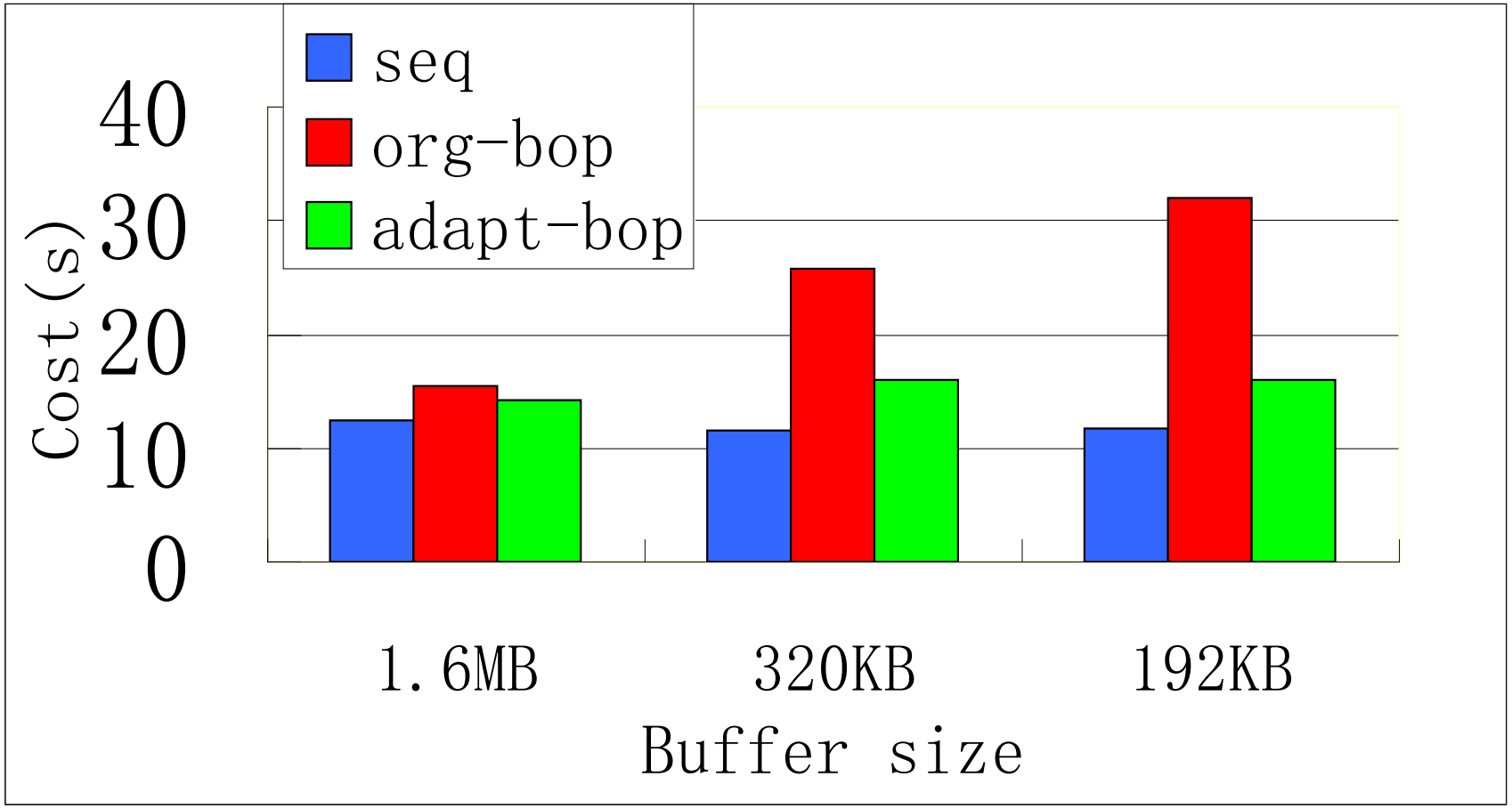
→Accuracy = 85.6%

Computation Efficiency

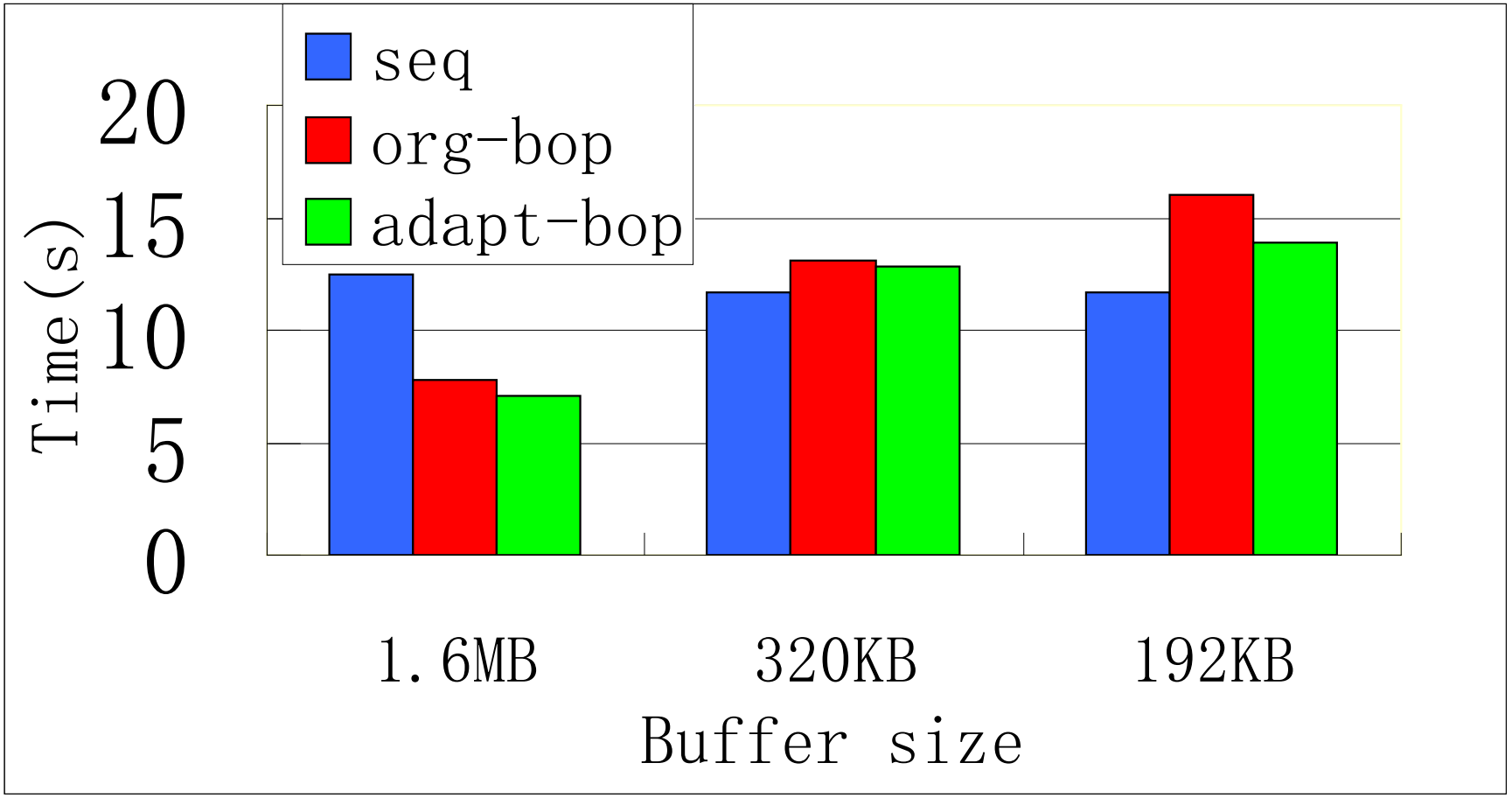


- Machine: Intel Pentium-D dual-core processors
- Compiler: gcc4.1
- Benchmarks
  - Gzip, Parser, Reduction
- Metrics
  - Cost
    - Total running time of all the processes
  - Time
    - Finishing time of a program

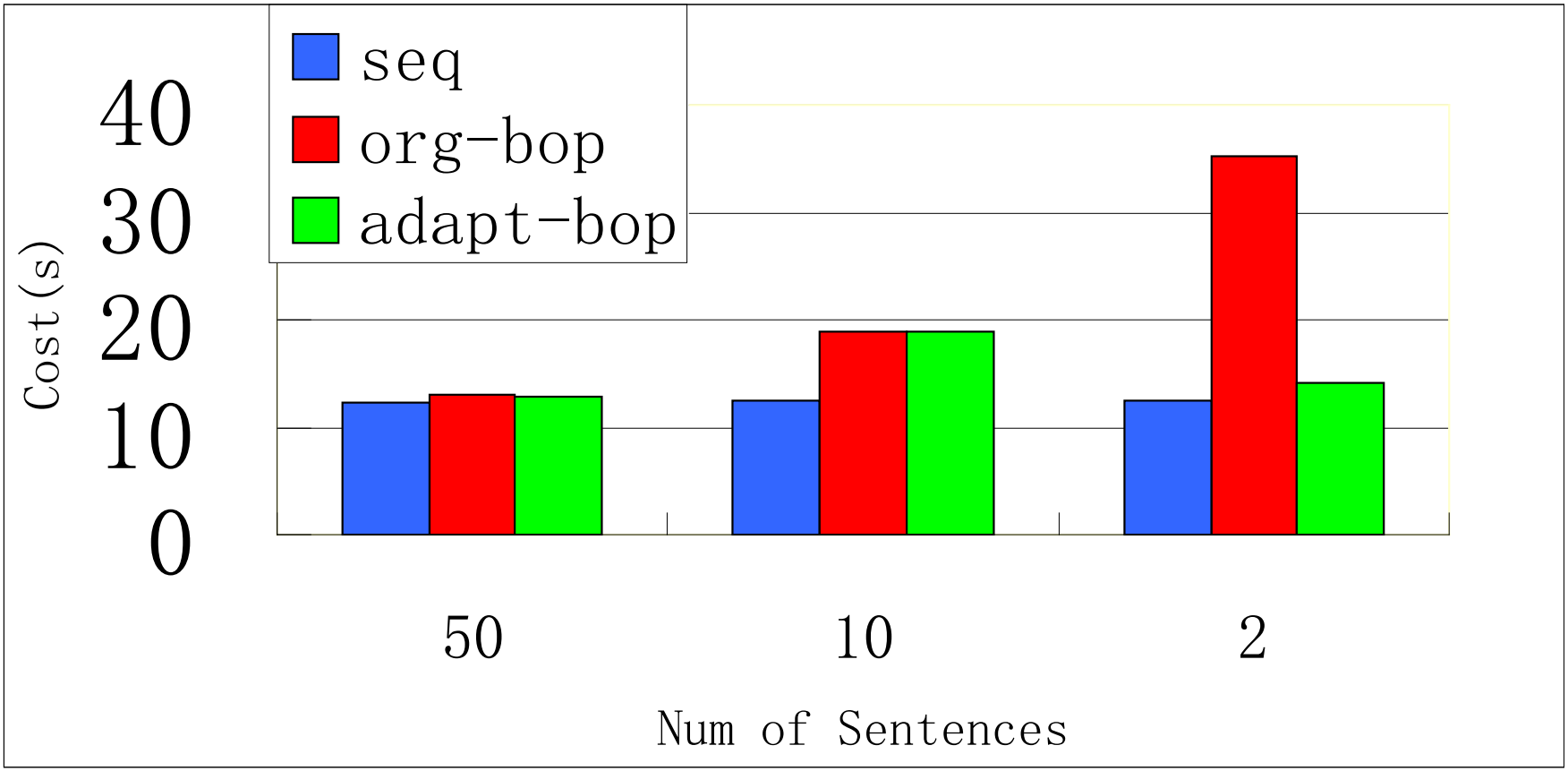
Efficiency comparison on gzip



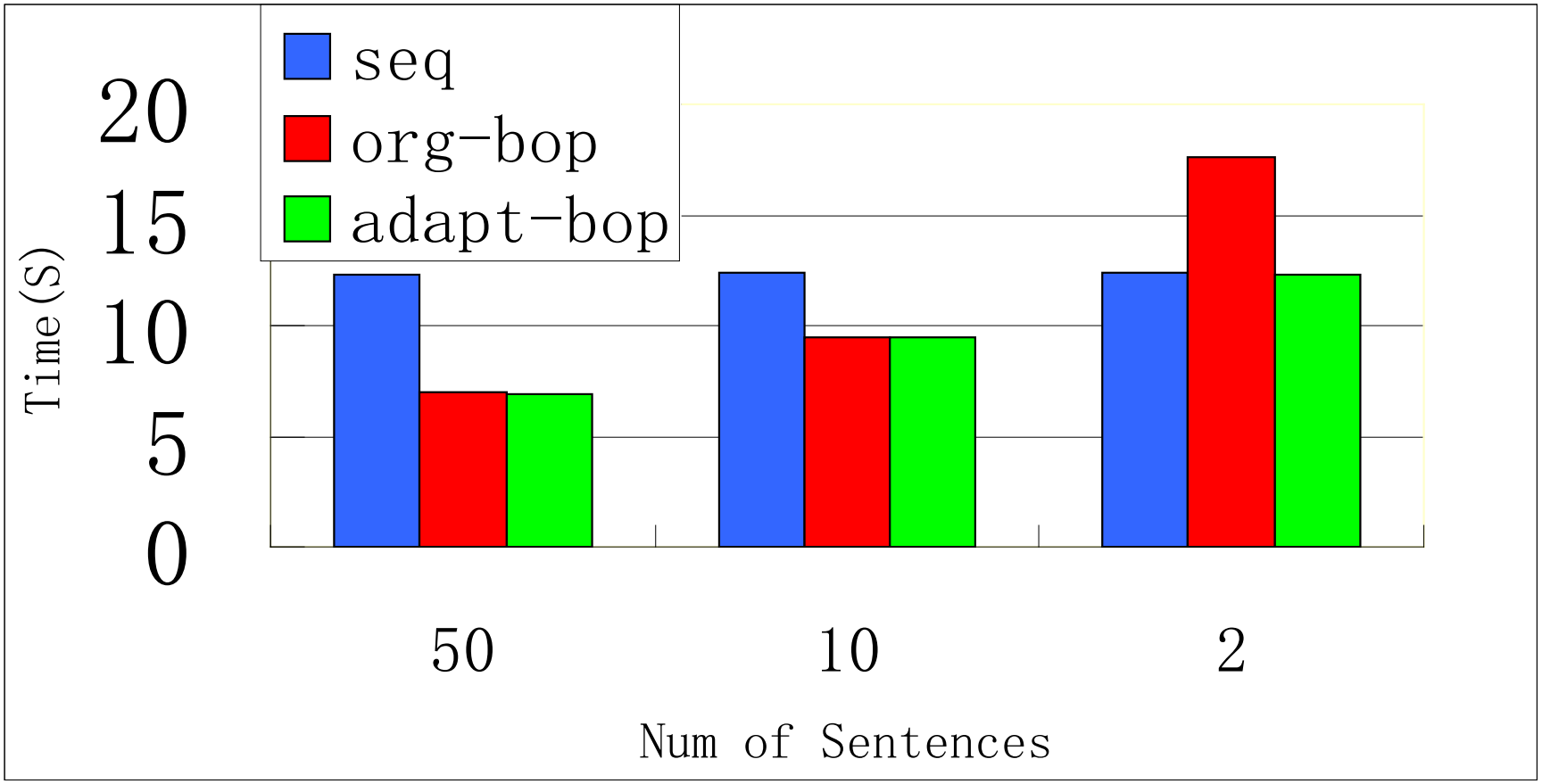
Efficiency comparison on gzip



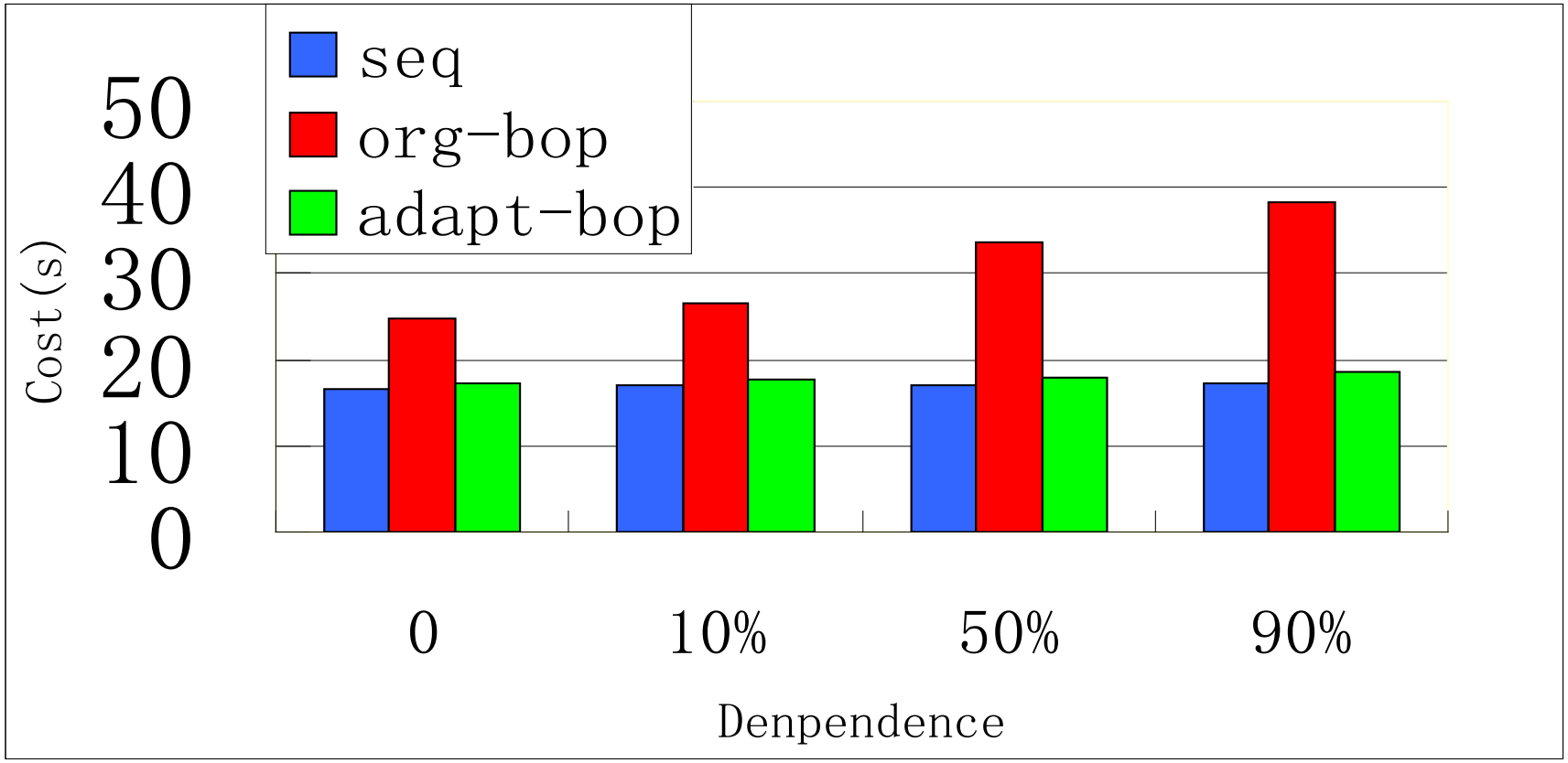
Efficiency comparison on parser



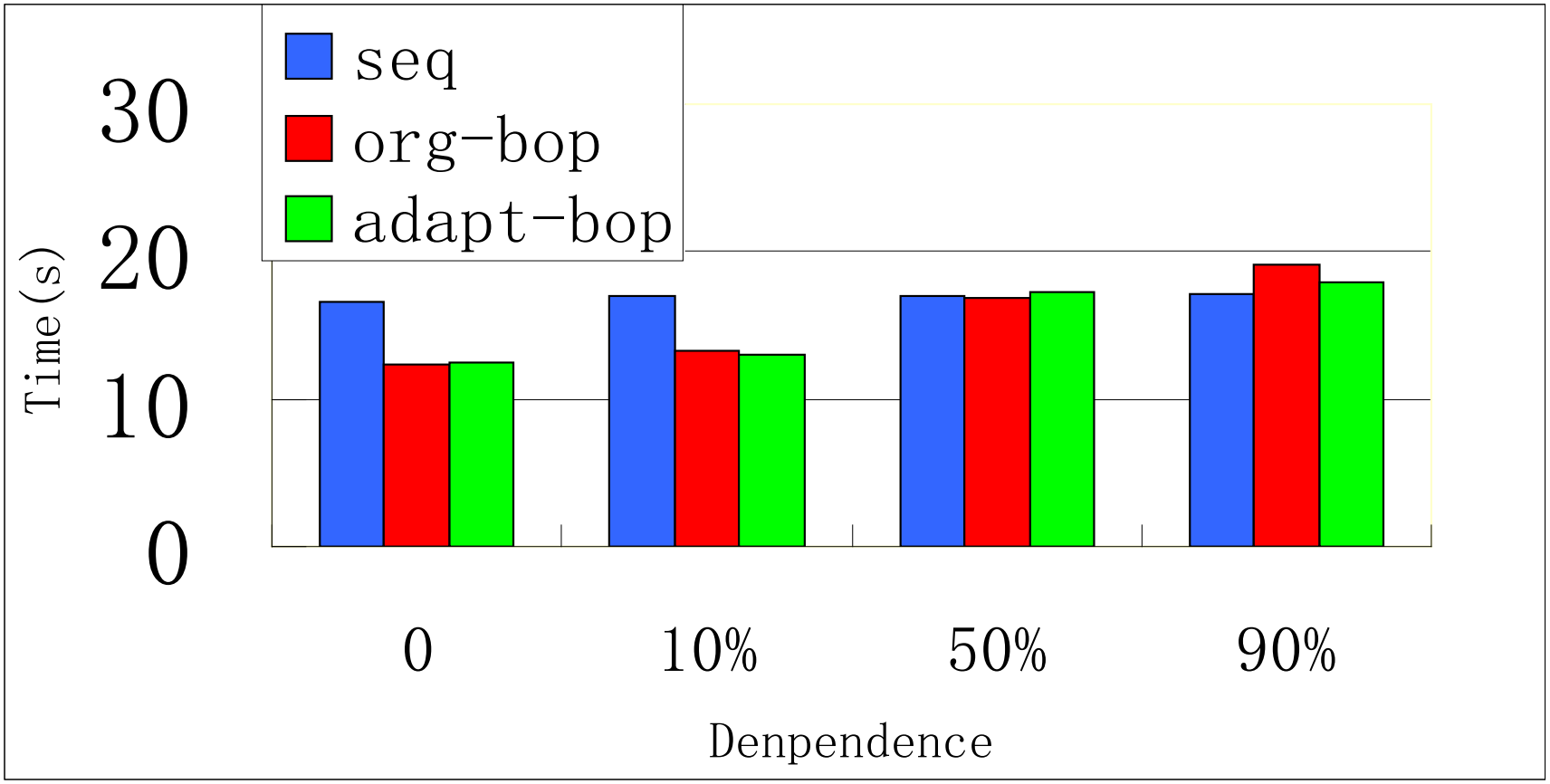
Efficiency comparison on parser



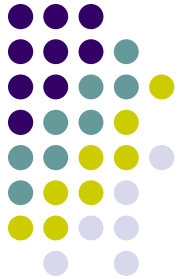
Efficiency comparison on Reduction



Efficiency comparison on Reduction



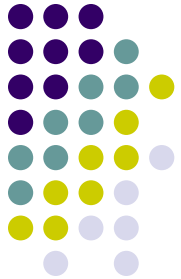
Outline



- Introduction to BOP
- Adaptive Algorithms
- Experimental Result
- Conclusion

Conclusions

- Failed Speculation is a problem
- Two adaptive algorithms
  - Last-value-based prediction
  - Decayed-history-based prediction
- Performance
  - High accurate prediction
  - Keep fast running speed
  - Reduce cost



Thanks!

Questions?