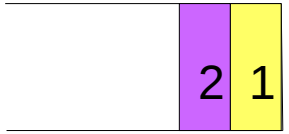
Sprint Ability: How Well Does Your Software Exploit Bursts in Processing Capacity?

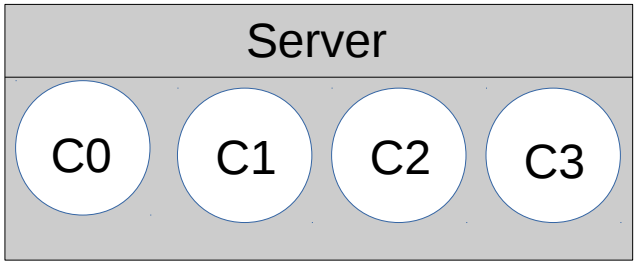
Nathaniel Morris*, Siva Meenakshi Renganathan, Christopher Stewart: The Ohio State University
Robert Birke and Lydia Chen: IBM

The Problem

- A request processing system
  - Google, Siri,Netflix rec.



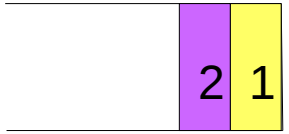
- A server with 4 cores
- Each core has 2 states
  - Slow (white) 1J per request
  - Fast (red) 2J per request
- System can only support 40J over all requests



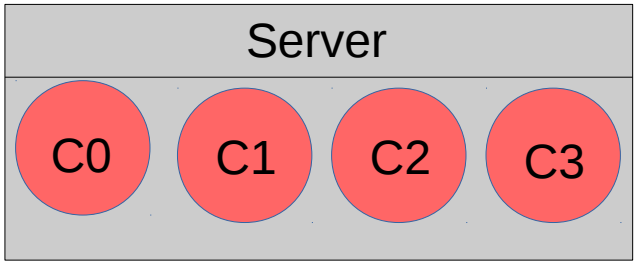
- Which configuration to use for each request?

The Problem

- A request processing system
  - Google, Siri,Netflix rec.



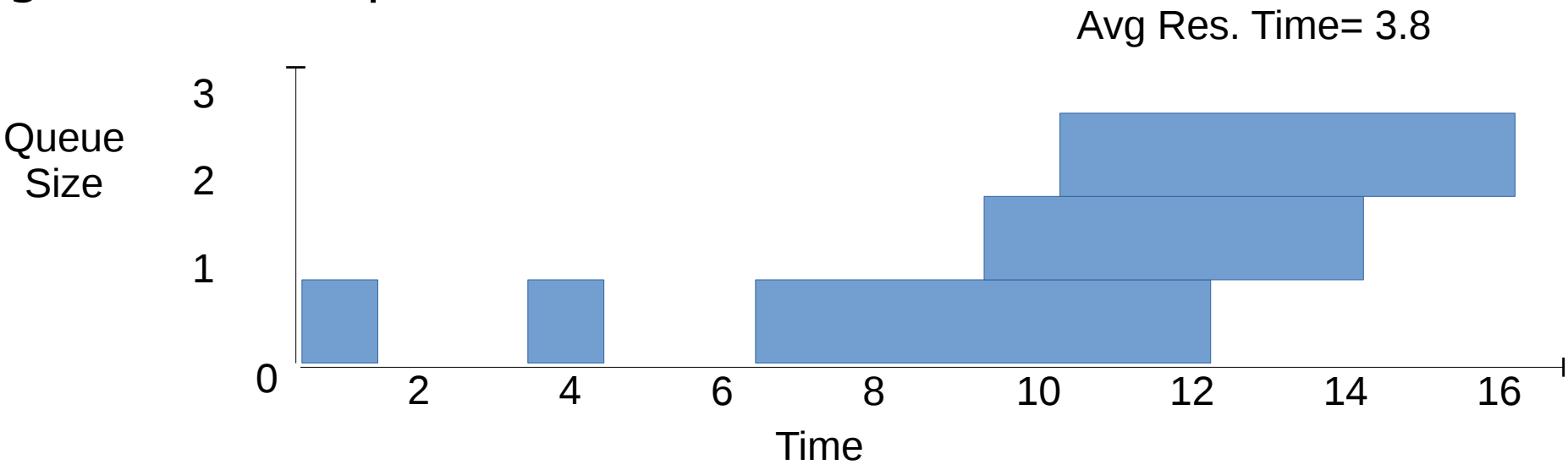
- A server with 4 cores
- Each core has 2 states
  - Slow (white) 1J per request
  - Fast (red) 2J per request
- System can only support 40J over all requests



- Which configuration to use for each request?

Motivating Example

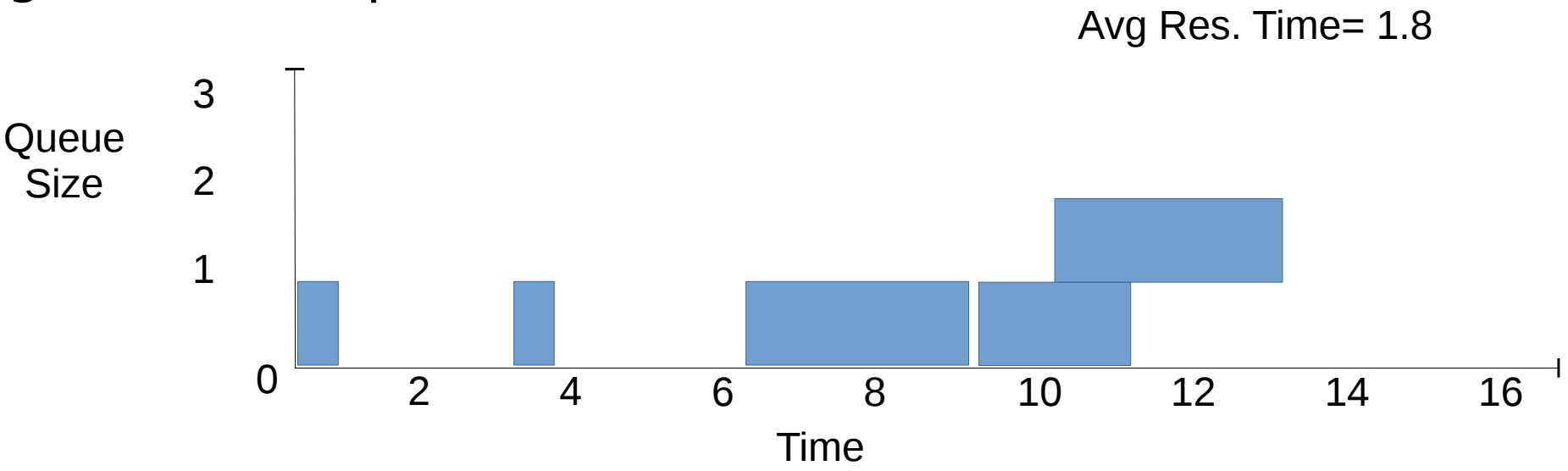
- Deciding which requests gets the fast processor



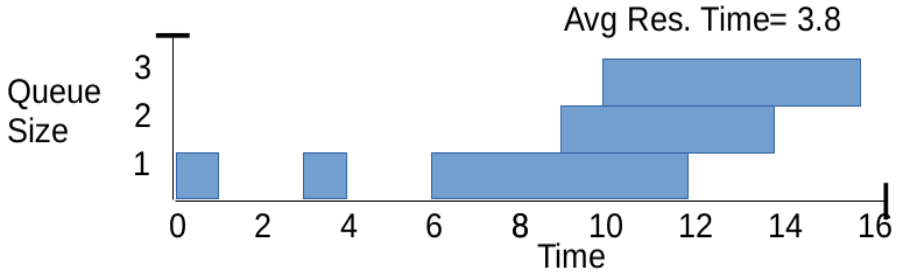
- Use only slow cores

Motivating Example

- Deciding which requests gets the fast processor

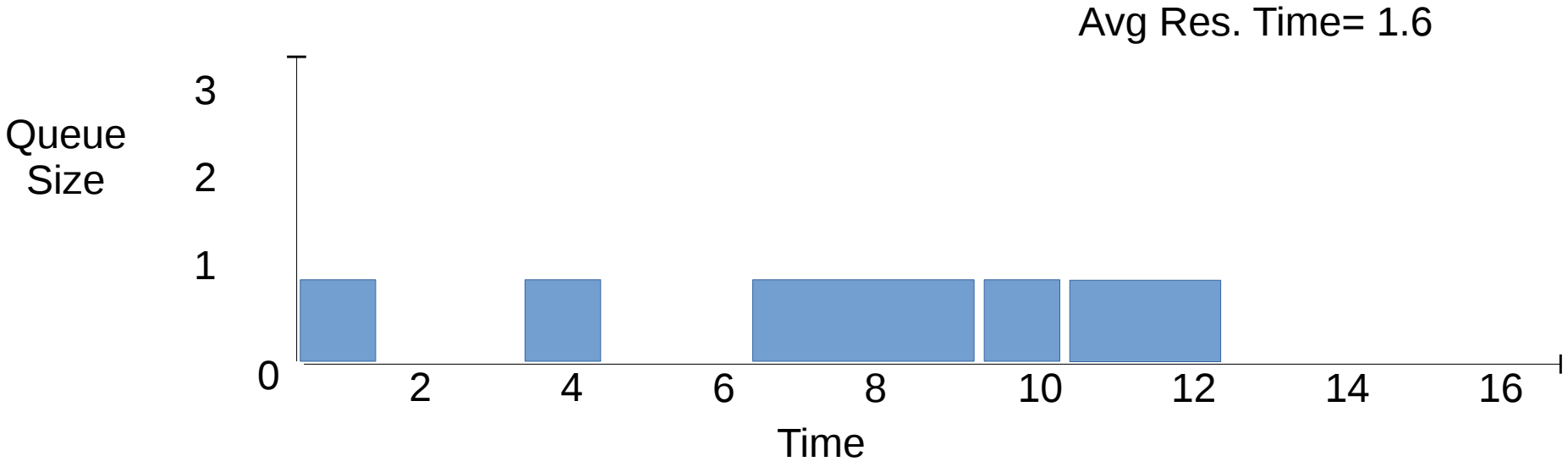


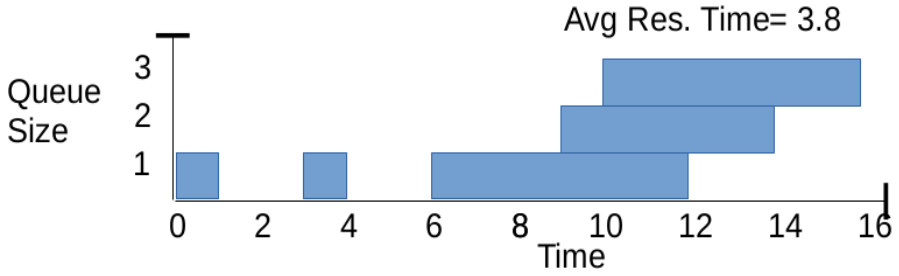
- Use energy budget on first requests (too greedy)



Motivating Example

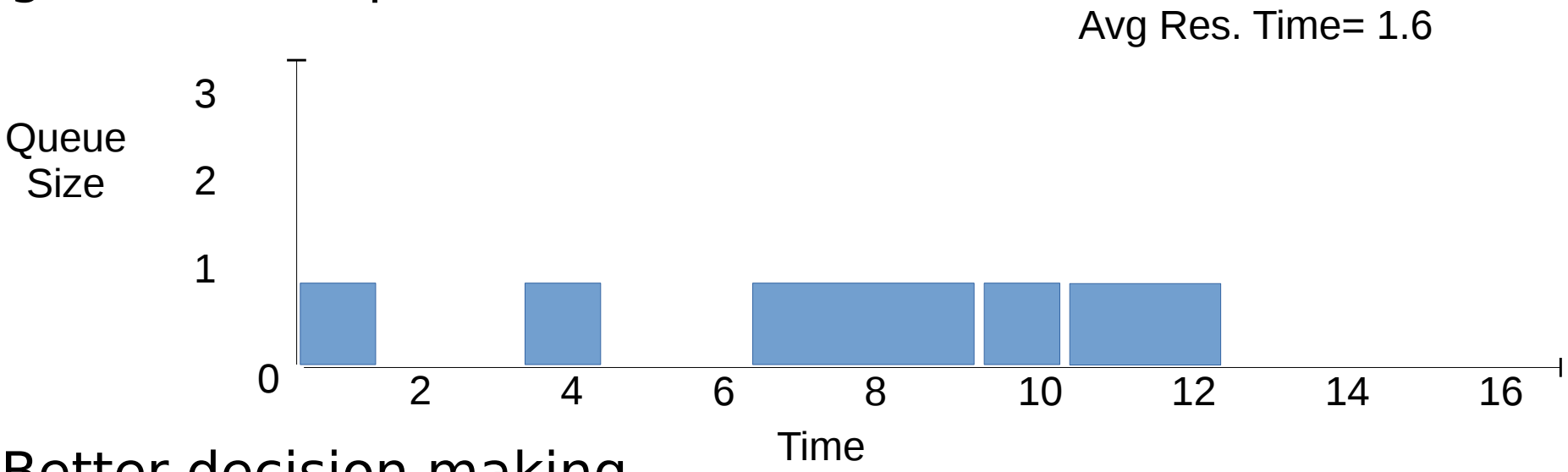
- Deciding which requests gets the fast processor





Motivating Example

- Deciding which requests gets the fast processor



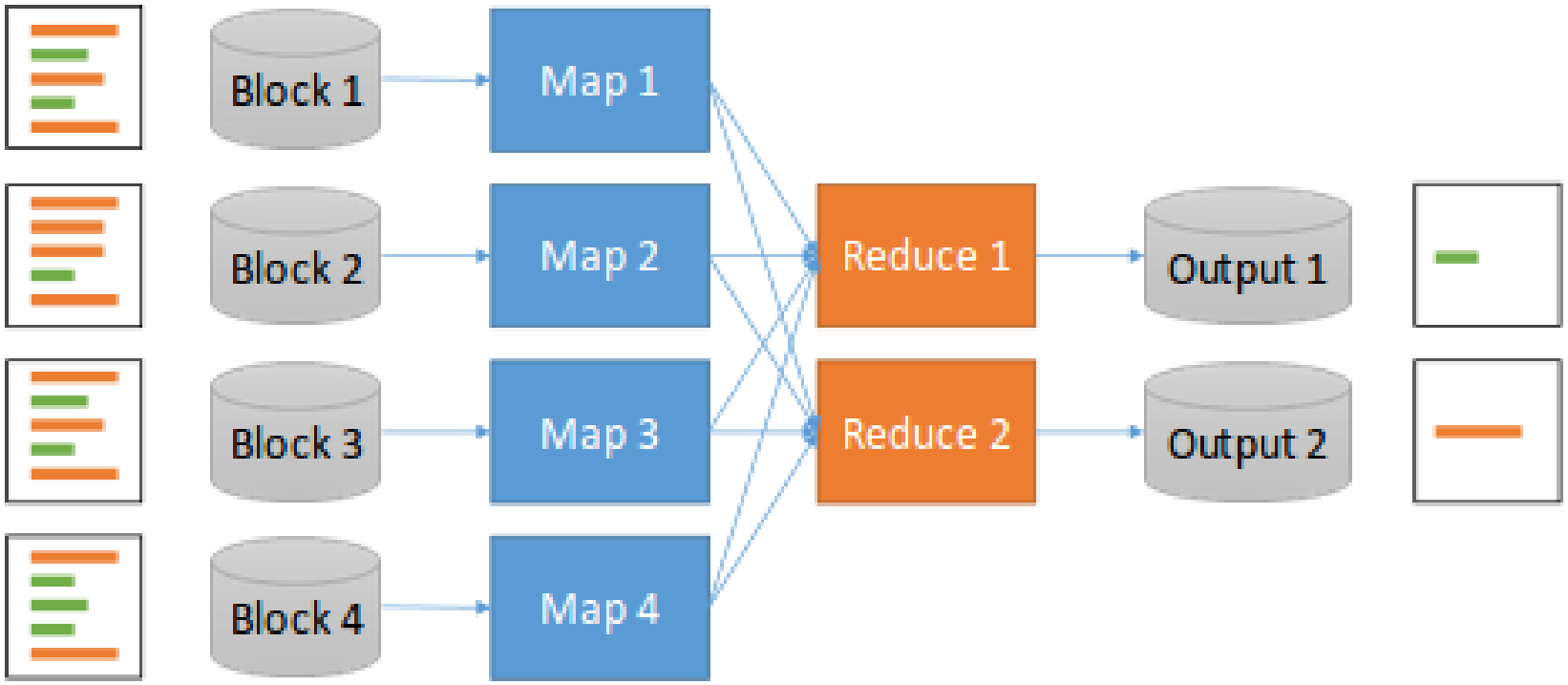
- Better decision making
- 2 challenges
  - Detect the moments which provide greatest utility for sprinting
  - Preserve budget for those moments

Computational Sprinting

- Queries can be processed faster
  - Overclocking
  - Exceeding Circuit Breaker Capacity
  - Return lower answer quality
- Any of these solutions can be harmful
  - However, they can be used safely in short bursts
  -

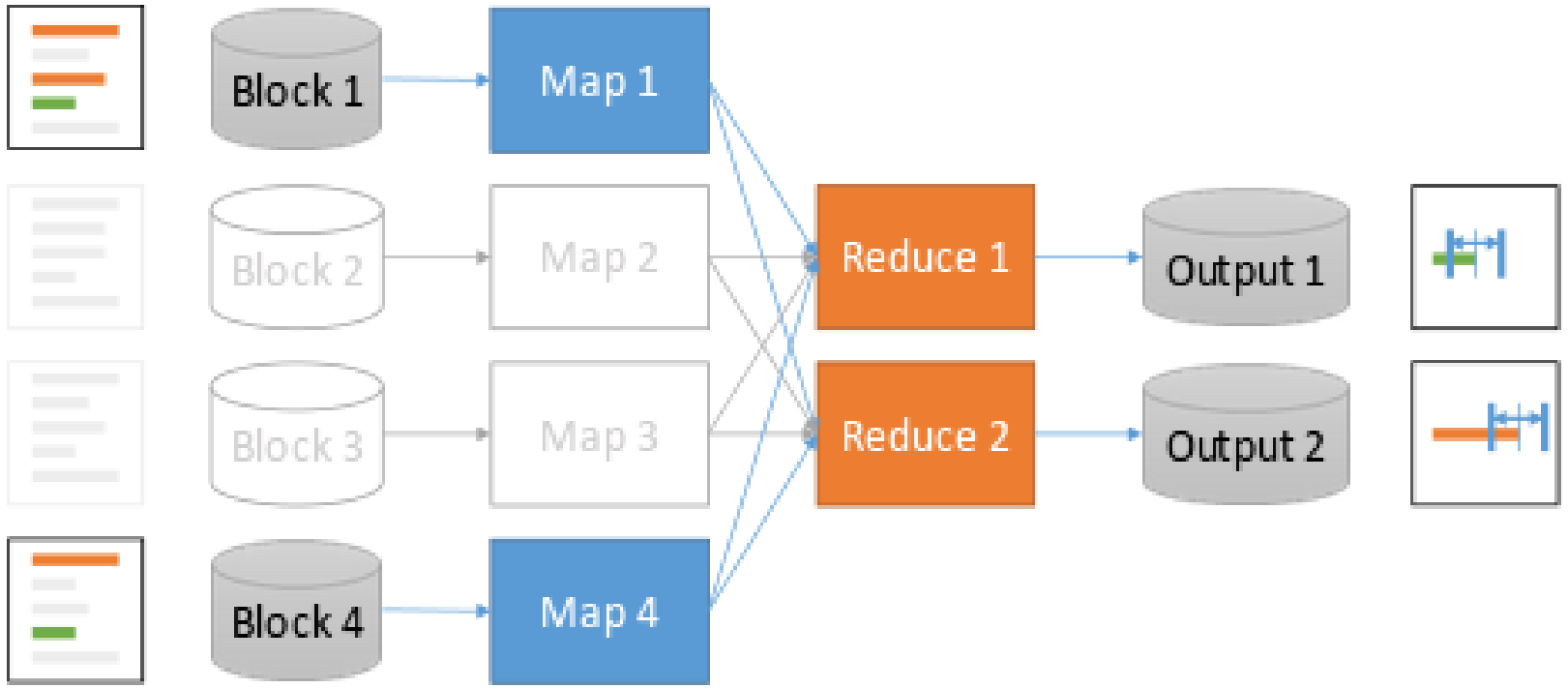
Examples of Sprinting

- ApproxHadoop



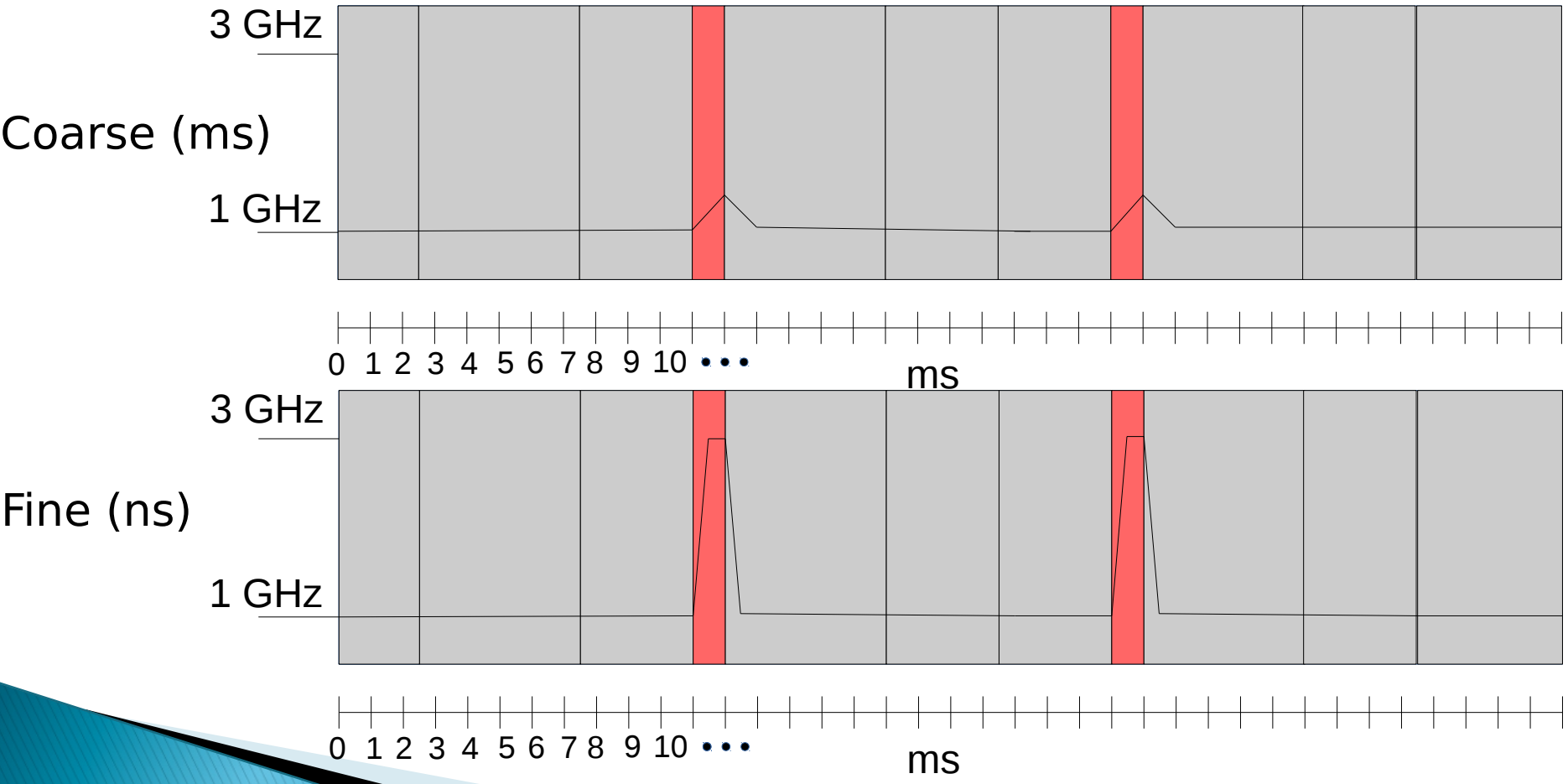
Examples of Sprinting

- ApproxHadoop



Examples of Sprinting

- Adrenaline



Benefits of Sprinting

- Speed up queries that demand heavy computation
- Subdue workload surges
  - Prevent queuing delays
- Improve response time
  - Can improve response time if triggered when needed
  - We need sprinting policies
- Sprinting policies manage the mechanisms
  - Typically policies with sprinting perform better

Our Contribution

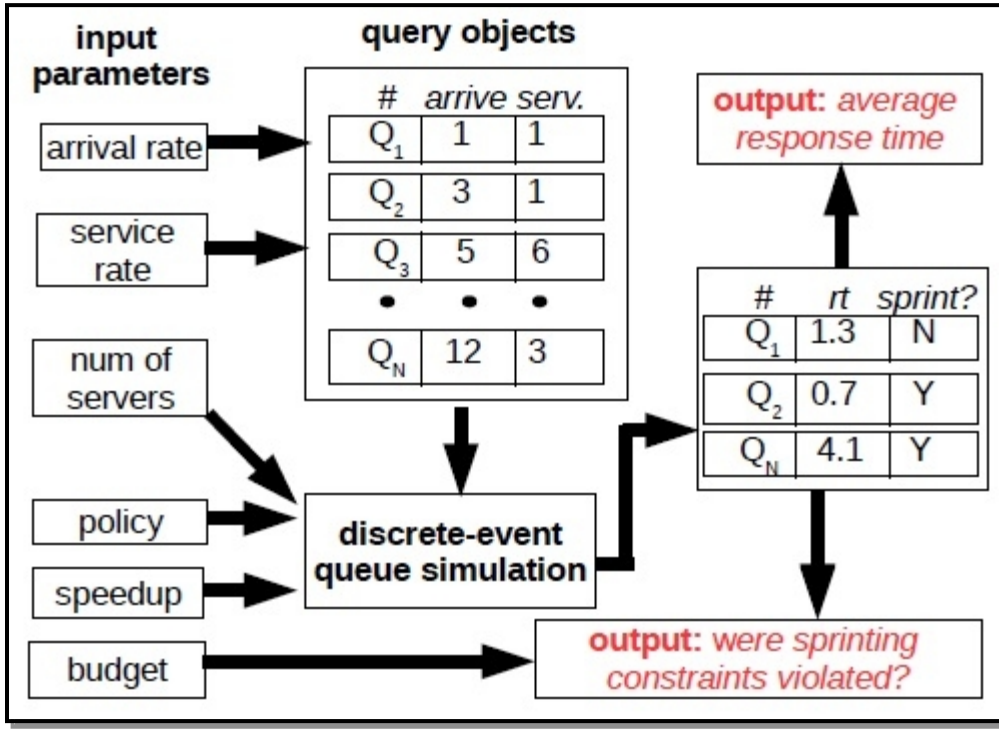
- We propose Sprint Ability
  - A metric that assesses sprinting policies
  - Divide performance achieved by the best performance under any sprinting policy
- Designed sprinting-aware simulator
  - Used to explore large parameter space to find optimal sprinting policy
  - reroutlab.org/projects.html

Sprint Ability (SA)

- Formal Definition
  - $$SA = \frac{Perf\left(P_{trg}, \vec{c}\right)}{Perf\left(P_{opt}, \vec{c}\right)}$$
- Only one sprinting policy can be used at a time
  - Empirical approaches measure performance directly to rank policies
  - This approach is time consuming
    - Adjustable parameters
    - Many competing policies
- Our Vision
  - P_{opt} is obtained from models or simulation
  - SA does not replace first-class metrics

A Simulator

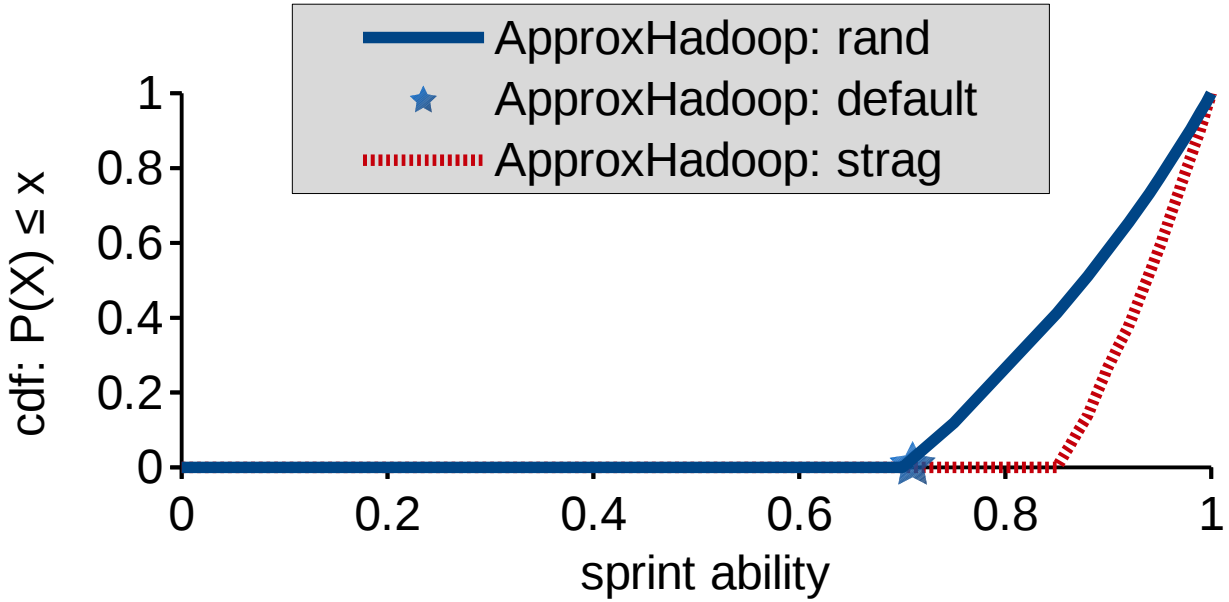
- M/M/k simulator
  - Supports wide range of policies (arrival, departure, etc.)
  - Target specific type of queries (tail response time)
- Limitations
  - Simulation slower than models
  - Models are hard when service rate depends on queue size



Our Questions

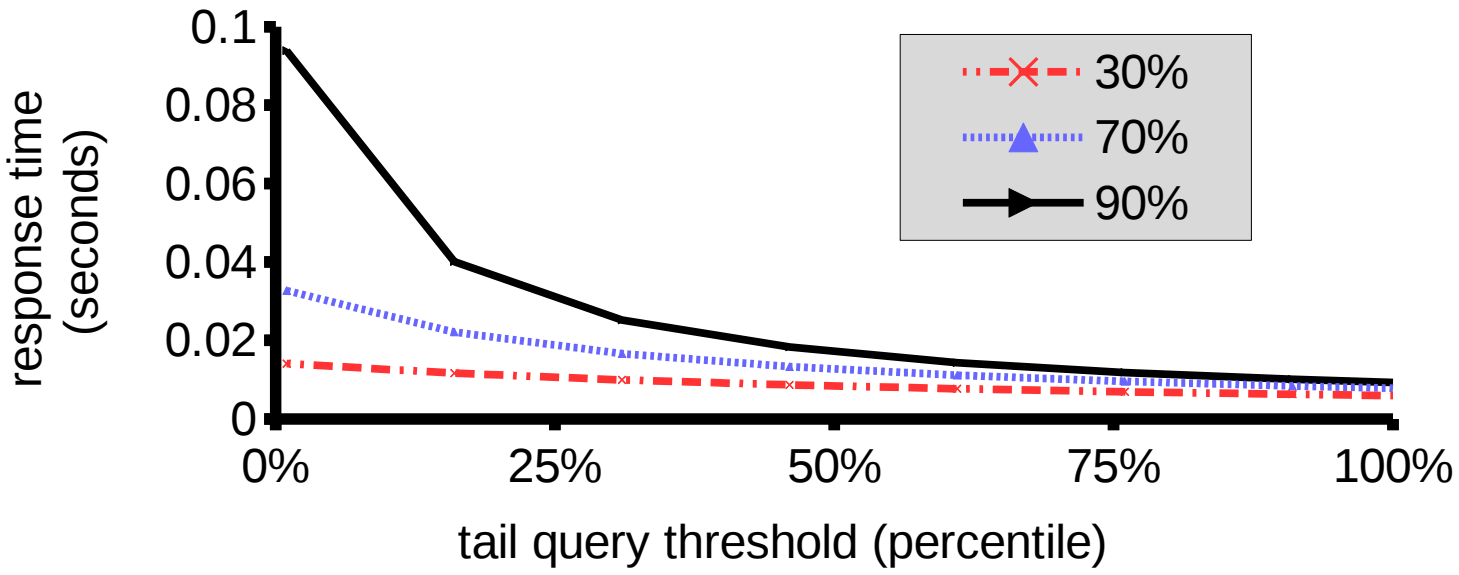
- How much better are the best sprinting policies?
- Do good sprinting polices share conspicuous features?
- Are the best sprinting policies affected by factors that change?
  - Failures, arrival rates, renewable energy

Early Results



- ApproxHadoop
  - 2 mechanisms: random and straggler dropping
    - 71% SA - default
    - 97% SA – straggler

Tail Query Threshold



- Speedup of Adrenaline under fixed speedup
  - Increasing tail threshold reduces response time
- System Utilization
  - Higher utilization granted greater performance gains

Resource Management

- SA as a variable in an autonomous system
  - Adaptive sprinting policies
    - Mixed workloads
  - Minimizing average queuing delay
    - Effectively sprinting when needed
- SA helps with resource allocation
  - All cores may not need to be overclocked

Conclusion

- A new era in which processing capacity is available in short bursts (aka sprinting)
- New techniques and metrics are needed to lower response time
- Sprint Ability complements existing metrics
  - Distinguishes policy from mechanism
  - Should be a part of resource management’s tools
- We built a simulator to evaluate Sprint Ability
  - Good policies are not always the best!