THE OHIO STATE UNIVERSITY

MURI Meeting
October 7, 2014

Analysis and Design of Complex Systems with Multivariate Heavy Tail Phenomena

Ness B. Shroff

Depts. ECE & CSE, The Ohio State University

E-mail: shroff@ece.osu.edu

Collaborators Yoora Kim, Irem Koprulu, R. Srikant, and Y. Zheng

Four Research Directions

Direction I Analyzing statistical metrics of different mobility models (Lévy, Explore return model, etc.): first exit time, contact time…

Direction II Exploiting opportunities (node mobility, channel variation, predictibility) for resource allocation in wireless nets.

Direction III Modeling and Analyzing Influence Propagation on Evolving Social Networks

Direction IV Developing scheduling algorithms for data center/ cloud computing systems

Progress Overview

Direction I Analyzing statistical metrics of different mobility models (Lévy, Explore return model, etc.): first exit time, contact time, intercontact time…

- First exit time analysis for Lévy flight model in $\mathbf{R}^N$ [Advances in Applied Probability, 2015]
  - Due to the heavy-tailed jump-length, the sample path consists of many short jumps and occasional long jumps --- N-variate heavy tailed distribution in $\mathbf{R}^N$
- Extension to Explore/Return model for more detailed human mobility modeling (ongoing work)
- Extension to directional drift model (ongoing work)
- Extension to Contact Time Analysis (ongoing work)

Progress Overview

Direction II Exploiting various opportunities (node mobility, channel variation, user predictibility) for resource allocation in wireless networks

- Optimal scheduler design for content sharing [INFOCOM'13]
  - Developed schedulers that maximized the system utility for sharing content, subject to hard deadline constraints
- Design of wireless data off-loading schemes [Mobihoc 2014]
  - A coupled queueing problem with bi-variate heavy tailed on/off service time distribution
  - Analysis of the reneging probability (probability that primary network is not chosen)
  - Expected waiting time analysis
  - Asymptotic study (ongoing work)

Progress Overview

Direction III Modeling and Analyzing Influence Propagation on Evolving Social Networks

- Modeled and analyzed the spread of multiple competing opinions (influences) in an evolving social network
  - An opinion/movement starts with an initial number of followers
  - Newcomers are added according to a hybrid of the Preferential Attachment (PA) model (nodes with higher degrees attract more connections than nodes with lower degrees) and the Random Attachment model (new nodes connect to the existing nodes uniformly).
- Derived the time evolution of the expected number of people adopting different opinions as a function of (i) the initial number and connectivity of the propagators of each opinion and (ii) the persuasion power or strength of each opinion
- Ongoing work:
  - Personalized attachment and influence dynamics that consider newcomers preexisting affinities.
  - Studying the multivariate degree distributions of opinion sub-networks.

Progress Overview (cont'd)

THE OHIO STATE UNIVERSITY

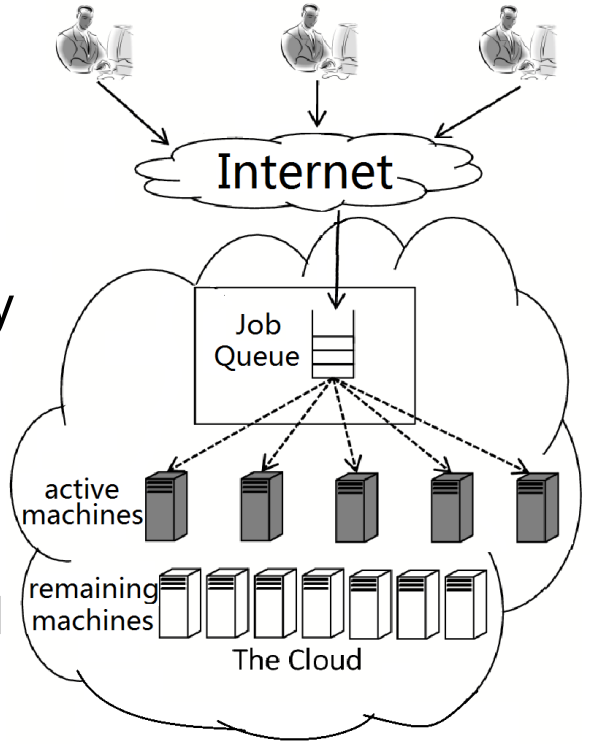
Direction IV Developing scheduling algorithms for data center/cloud computing systems

- Analysis and design of MaP/Reduce type scheduling algorithms with multi-variate heavy tailed dependency for minimizing the total flow time in the system
- Prove that the flow time minimization problem is strongly NP-hard and does not yield a finite competitive ratio [IEEE INFOCOM'13]
- Developed 2-approximation probabilistic competitive ratio pre-emptive scheduler that is independent of the nature of job size distributions [IEEE INFOCOM'13]
- Low-latency algorithms in the large-system limit for both pre-emptive and non-pre-emptive schedulers [submitted]
- Coding & Queueing: Optimal-Latency Data Retrieving in Storage Clouds [IEEE INFOCOM'13, & more recent work submitted]

Direction IV Developing scheduling algorithms for data center/cloud computing systems

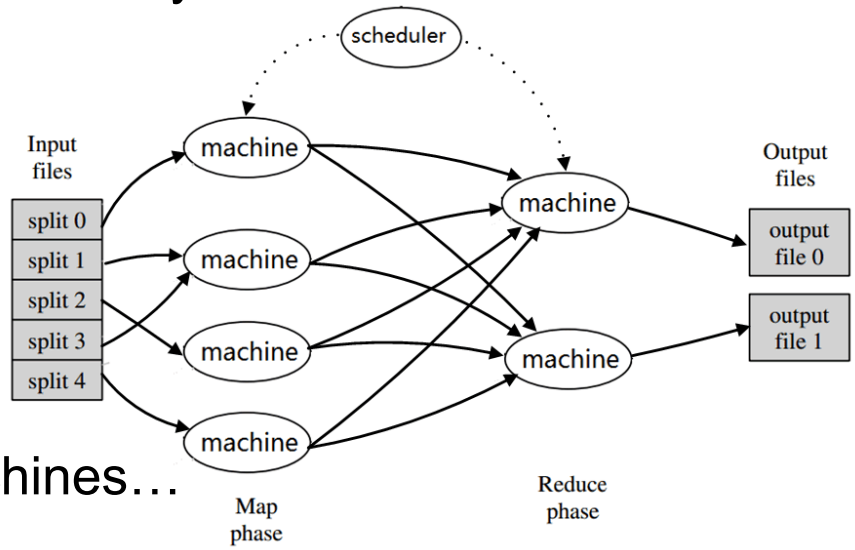
Data Centers/Cloud Systems

- Data Centers
  - Facility containing a large numbers of machines
    - Has roots in huge computer rooms of the early ages!
  - Services: IaaS, PaaS, SaaS
  - They process very large datasets
    - Use a (variation of) programming model called Map Reduce
      - Developed (popularized) by Google
      - Easy to use/program/make scalable (without extensive training)
      - Nearly ubiquitous (Google, IBM, Facebook, Microsoft, Yahoo, Amazon, eBay, twitter…)
      - Used in a variety of different applications (distributed grep, sort, AI, scientific computation, image processing, …)



MapReduce

- Consists of two elemental processes
  - Each arriving job undergoes a Map and a Reduce Phase
- Map phase:
  - Takes an input and divides into many small sub-problems (tasks)
  - Operations can run in parallel on potentially different machines
- Reduce phase(s)
  - Combines the output of Map
  - Usually occurs after the Map phase is completed
    - **Phase precedence constraint**
    - **Multiple reduce phases**
  - Operations can run on parallel machines…



Goal: Schedule these Map and Reduce tasks in order to minimize the total flow time in the system

Flow Time

- Amount of time a job spends in the system
  - Includes both waiting and processing time of all the phases of a job
  - Important metric of performance
- Key challenge: To minimize the flow time, scheduling decisions need to maintain phase precedence constraints.

Relationship to Project

THE OHIO STATE UNIVERSITY

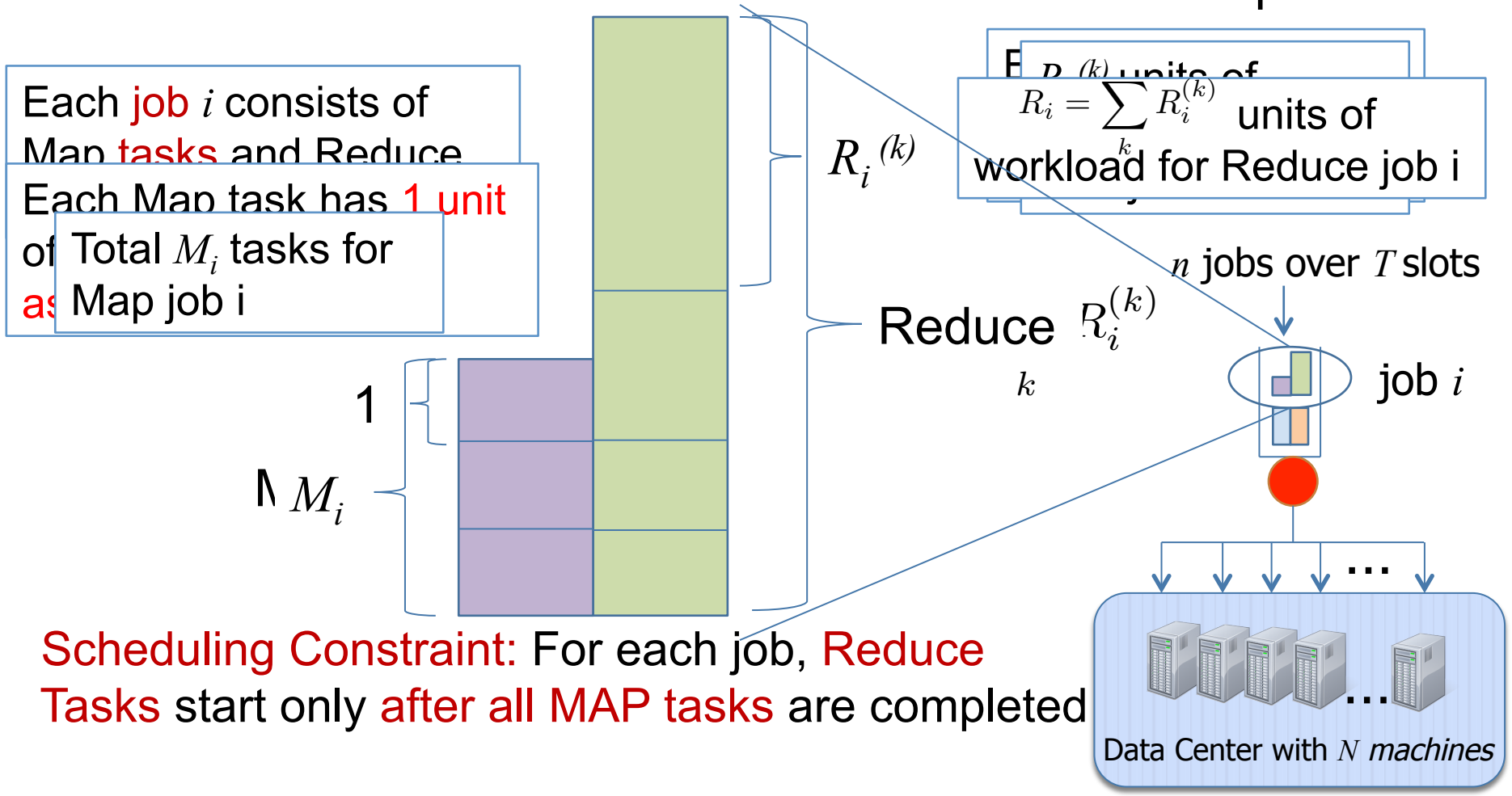
- Number of Map tasks in a job is heavy tailed (or truncated heavy tailed)
- Size of each reduce task is heavy tailed (or truncated heavy tailed)
- Multiple phases within a job are dependent (e.g., MAP, Reduce1, Reduce 2, Reduce 3…).
  - Random vector of task workload ~ Multivariate heavy tail

Sample References:

- Jian Tan, Xiaoqiao Meng ; Li Zhang, "Coupling Task Progress for MapReduce Resource-Aware Scheduling", IEEE INFOCOM 2013.
- G. Ananthanarayanan, A. Ghodsi, A. Wang, D. Borthakur, S. Kandula, S. Shenker, and I. Stoica. PACMan: coordinated memory caching for parallel jobs. In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, NSDI'12, San Jose, CA, 2012. USENIX Association.
- S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan. An analysis of traces from a production MapReduce cluster. In Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID '10, pages 94– 103, Washington, DC, USA, 2010.

Model

- Time is slotted
- *N* "Machines": Each machine runs 1 unit of workload per slot

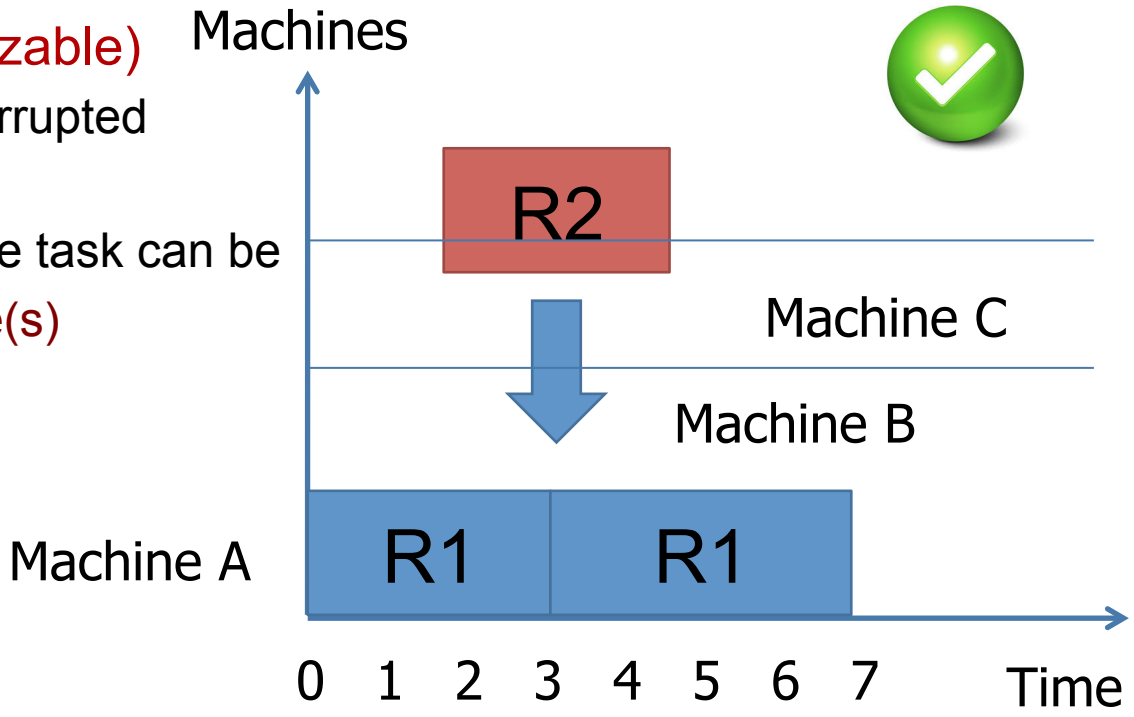


- Scheduling Constraint: For each job, Reduce Tasks start only after all MAP tasks are completed

Types of Schedulers: Treatment of Reduce Tasks

THE OHIO STATE UNIVERSITY

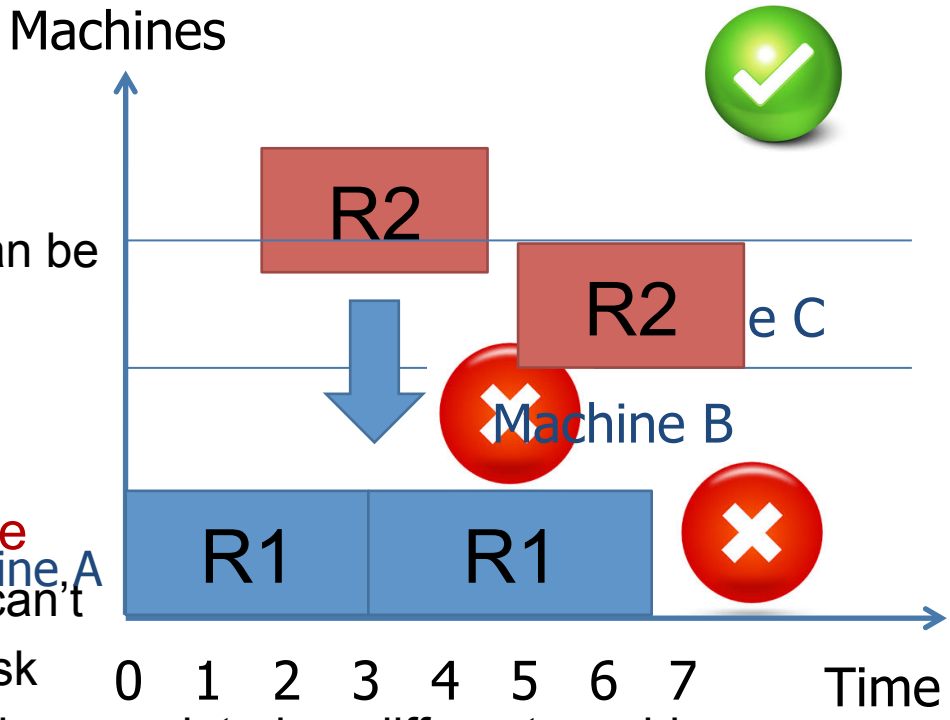
- Preemptive (also parallelizable)
  - Reduce tasks can be interrupted at the end of a slot
  - Remaining workload in the task can be executed on any machine(s)
  - Reasonable if overhead of data-migration is small
- Non-preemptive



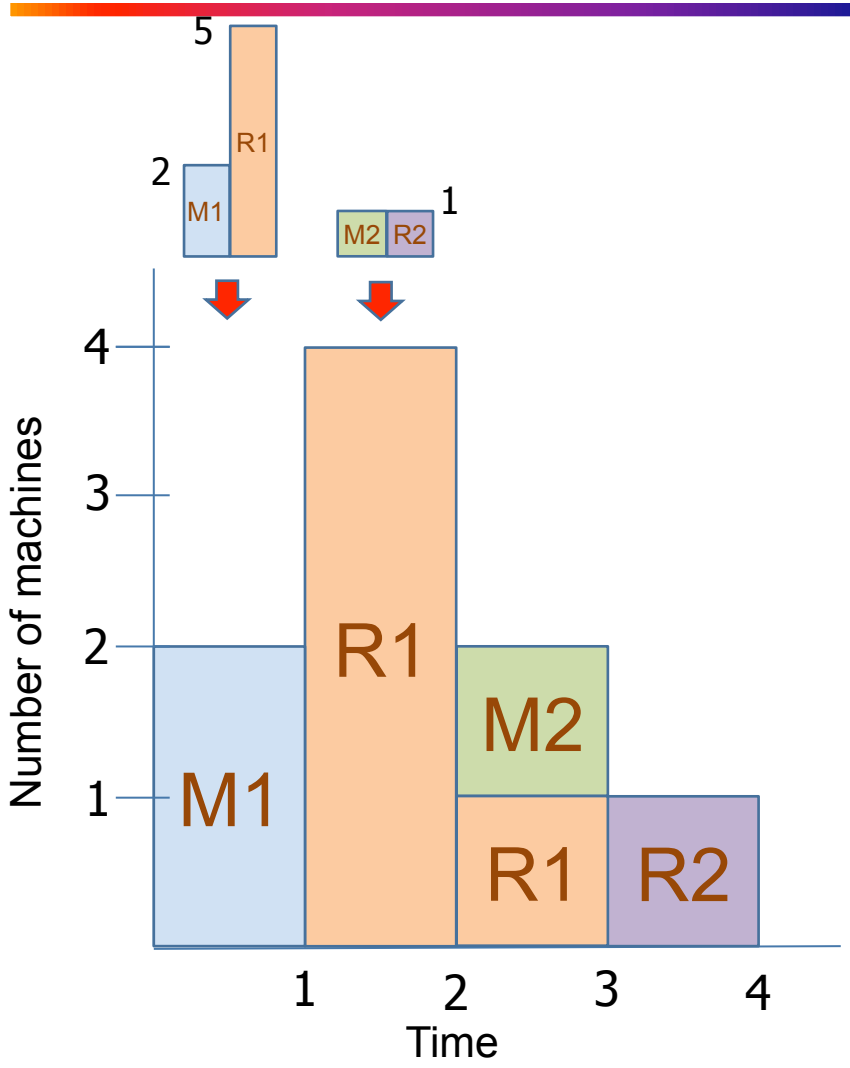
Types of Schedulers: Treatment of Reduce Tasks

THE OHIO STATE UNIVERSITY

- Preemptive/parallelizable
  - Reduce tasks can be interrupted at the end of a slot
  - Remaining workload in the task can be executed on any machine(s)
  - Reasonable if overhead of data-migration is small
- Non-preemptive/non-parallelizable
  - Once a Reduce task is started, it can't be interrupted till the end of the task
  - An individual Reduce task cannot be completed on different machines
  - But different tasks from same job can be assigned to different machines
- Note: Since Map tasks have unit workload, they cannot be interrupted.
  - In practice Map tasks may be > 1 unit, but small



MapReduce: FCFS scheduler



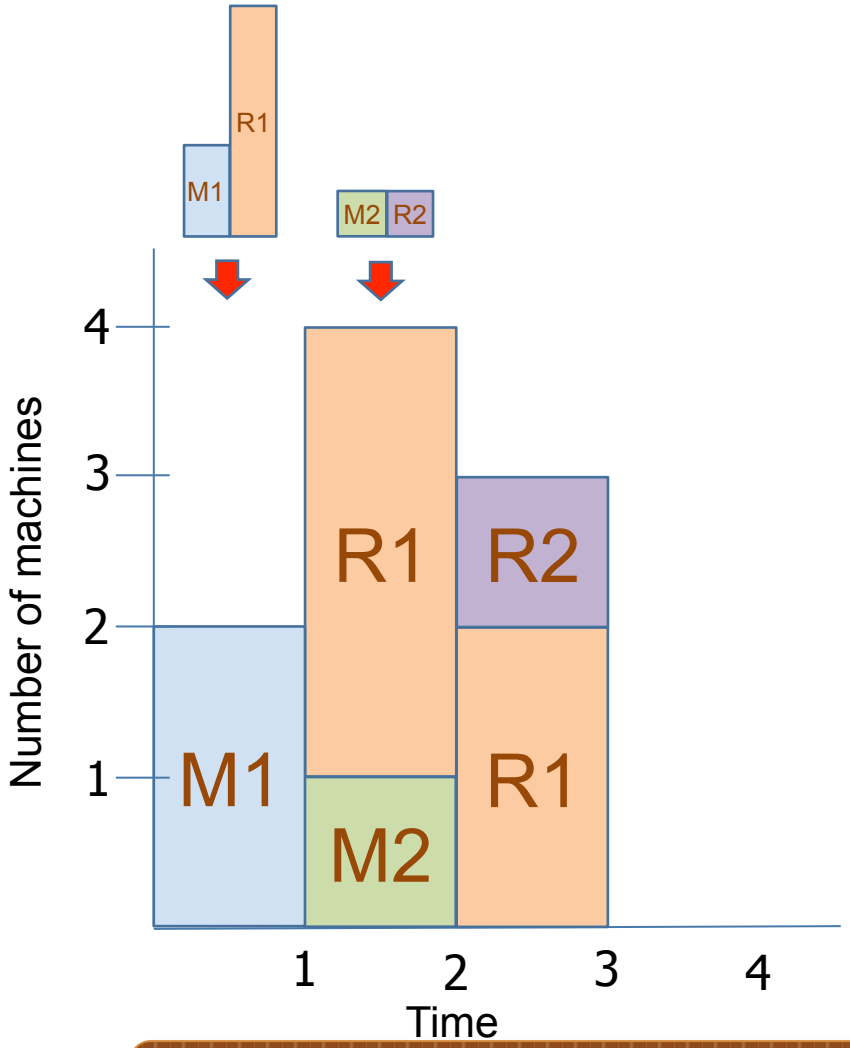
FCFS scheduler with
4 machines, 2 jobs

Map must finish before
Reduce can start

Flow-time of a job: time spent by job in the system

FT of Job 1: 3 time slots
FT of Job 2: 3 time slots
Total FT : 6 time slots

MapReduce: Smarter Scheduler



"Smarter Scheduler"

Flow-time: time in the system

FT of Job 1: 3 time slots

FT of Job 2: 2 time slots

Total FT: 5 time slots

Goal: Minimize total flow-time

Flow-Time Minimization Problem: Preemptive and Parallelizable Scenario

THE OHIO STATE UNIVERSITY

$$\min_{m_{i,t},r_{i,t}} \quad \sum_{i=1}^{n} \left(f_i^{(r)} - a_i + 1 \right)$$

Minimize flow-time

$$s.t. \quad \sum_{i=1}^{n} (m_{i,t} + r_{i,t}) \leq N, \ \forall t,$$

Total # machines is N

$$\sum_{t=a_i}^{f_i^{(m)}} m_{i,t} = M_i, \ m_{i,t} \geq 0, \ \forall i,$$

Total map workload is M_i for job i

$$\sum_{t=f_i^{(m)}+1}^{f_i^{(r)}} r_{i,t} = R_i, \ r_{i,t} \geq 0, \forall i.$$

Total reduce workload is R_i for job i

Scheduling Constraint

Flow-Time Minimization Problem
Non-Preemptive & Non-Parallelizable Scenario

THE OHIO STATE UNIVERSITY

$$\min_{m_{i,t}, r_{i,t}^{(k)}} \sum_{i=1}^{n} \left(f_i^{(r)} - a_i + 1 \right)$$

Minimize flow-time

$$s.t. \quad \sum_{i=1}^{n} (m_{i,t} + \sum_{k=1}^{K_i} r_{i,t}^{(k)}) \leq N, \ \forall t,$$

Total # machines is N

$$\sum_{t=a_i}^{f_i^{(m)}} m_{i,t} = M_i, \ m_{i,t} \geq 0, \ \forall i,$$

Total map workload is M_i for job i

$$\sum_{t=f_i^{(m)}+1}^{f_i^{(r)}} r_{i,t}^{(k)} = R_i^{(k)}, \ \forall i, \forall k \in \{1, ..., K_i\},$$

Total reduce workload is $R^{(k)}_i$ for task k in job i

$$r_{i,t}^{(k)} = 0 \text{ or } 1, \ r_{i,t}^{(k)} = 1 \text{ if } 0 < \sum_{s=0}^{t-1} r_{i,s}^{(k)} < R_i^{(k)}.$$

Non-preemptive/Non-parallelizable

Both Problems are NP-hard in the strong sense
[Zheng, Sinha, Shroff, INFOCOM 2013]

Competitive Ratio

- Θ : set of all possible schedulers (including non-causal)
- F^{ζ} : Total Flow time of scheduler ζ (sum of FT of all jobs)
- Define $F^* = \min_{\zeta \in \Theta} F^{\zeta}$

- Competitive ratio: A given online policy *S* is said to have a competitive ratio of *c* if for any arrival pattern

$$\frac{F^s}{F^*} \leq c$$

Theorem

For any constant c_0 and any online scheduler *S*, there are sequences of arrivals and workloads in the non-preemptive case, such that the competitive ratio c is greater than c_0

➔ No causal policy gives a bounded competitive ratio

Efficiency Ratio Analysis

- Efficiency ratio: A given online policy *S* is said to have an efficiency ratio of γ if (over some probability space of arrivals)

$$\limsup_{T\to\infty} \frac{F^S}{F^*} \leq \gamma \text{ with probability } 1$$

Theorem

For a Markov job arrival process, and bounded first and second moments of job workload:

All work-conserving schedulers have a bounded efficiency ratio, for both preemptive and non-preemptive scenarios.

Thus Far

- There exist no schedulers that have a bounded competitive ratio in non-preemptive scenarios
- All work conserving schedulers have a bounded efficiency ratio (γ) for both preemptive & non-preemptive scenarios
- But the performance of these schedulers could still be quite poor (γ could be very large)
- Key Question: Can we design provably efficient schedulers in the Map-Reduce paradigm (i.e., small γ)?

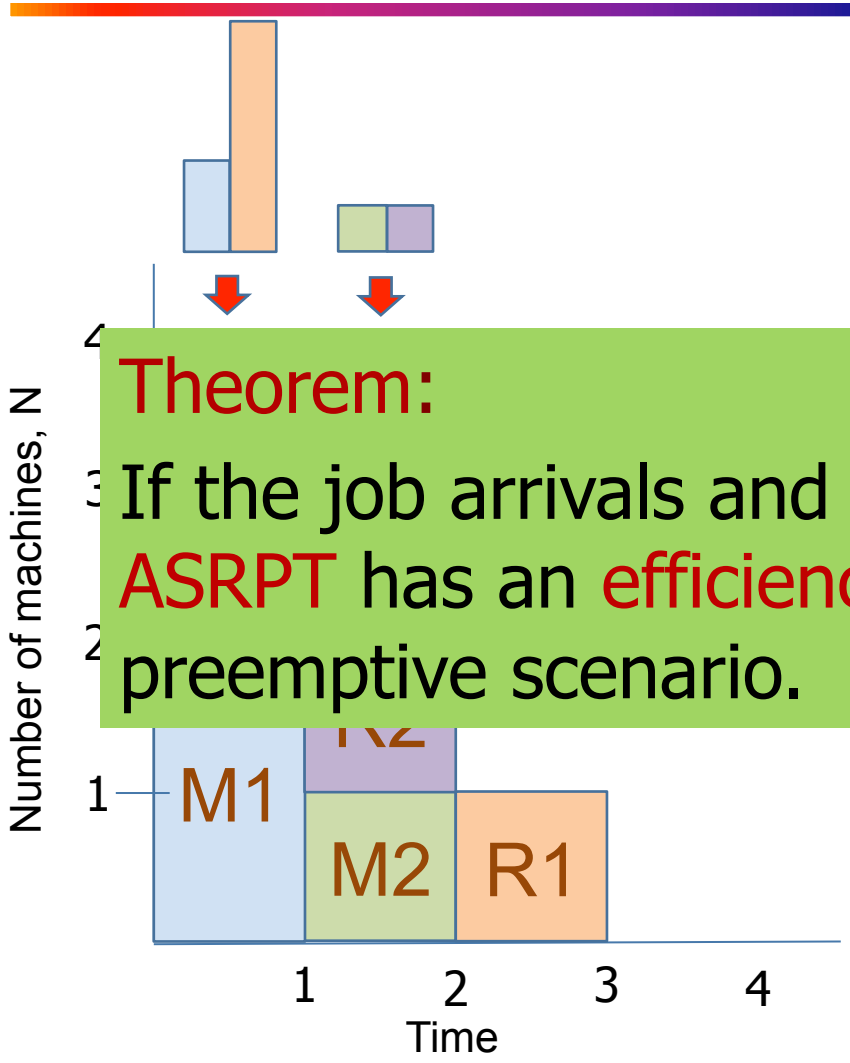
ASRPT: Available Shortest Remaining Processing Time [Quick Summary]

THE OHIO STATE UNIVERSITY

- Basic Approach:
  - An infeasible scheduler (SRPT)
    - Priority given to jobs with smallest remaining workload
    - In SRPT, Map and Reduce for a given job can be scheduled in same time-slot (infeasible in reality)
    - SRPT results in a lower flow time than any feasible (including non-causal) scheduler
  - ASRPT
    - Map tasks are scheduled according to SRPT (or sooner)
      - By running SRPT in a virtual manner
    - Reduce tasks greedily fill up the remaining machines
      - In the order determined by the shortest remaining processing time
  - Compare performance of ASRPT to SRPT to find γ

ASRPT: Available Shortest Remaining Processing Time

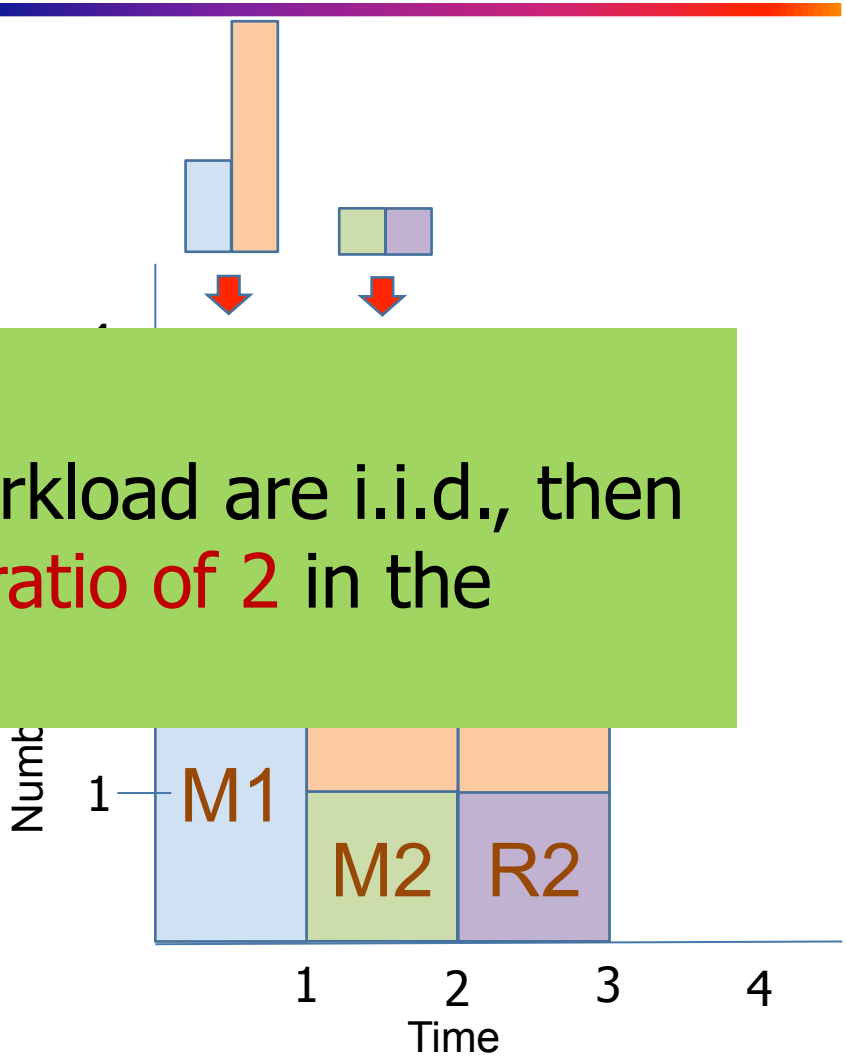
THE OHIO STATE UNIVERSITY



SRPT: Total Flow-time 4 units

Theorem:

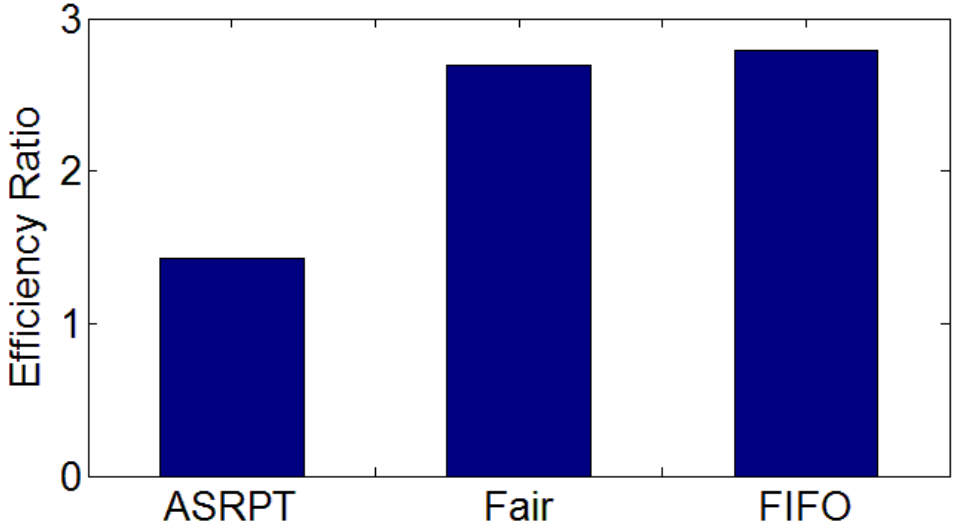
If the job arrivals and workload are i.i.d., then ASRPT has an efficiency ratio of 2 in the preemptive scenario.



ASRPT: Total Flow-time 5 units

Simulation

- N = 100 machines, total time slots T = 800
- # Reduce tasks in each job is 10
- Job arrival process: Poisson; λ = 2 jobs per time slot.
- Preemptive Scenario $Map : Exp(5)$ $Reduce : Exp(20)$
- Schedulers
  - ASRPT
  - Fair
  - FIFO
- ASRPT has smaller efficiency ratio than Fair and FIFO



- Similar results for dependent heavy tailed MAP/Reduce workloads and non-pre-emptive scenarios.

Simpler Schedulers

- ASRPT is simple but needs sorting (Complexity increases with the size of the system.)
- ASRPT requires prior knowledge of the amount of time needed to complete a job
  - Information may not be available in practice
- Question: Can we design even simpler schedulers?
- Yes
  - Approach: Exploit the fact that most data centers have a large number of machines to process jobs

A Large Number of Machines

THE OHIO STATE UNIVERSITY

- In a data center, there are many machines: $N \to \infty$
  - Design asymptotically optimal schedulers to minimize the flow time under both fixed $\rho < 1$ and heavy traffic $\rho \to 1$ regimes

- A schedule *S* is asymptotically optimal if

$$\lim_{N\to\infty} \frac{F^S(N,T)}{F^*(N,T)} = 1 \ w.p.1 \quad \text{for any given } 0 < T \leq \infty$$

where *T* is the total time slots, F^S is the total flow time of scheduler *S* (sum of FT of all jobs), and F^* is the minimum total flow time (in total time T) over all schedulers, including non-causal schedulers.

Asymptotic Optimality

- Main result: All work-conserving schedulers are asymptotically optimal in the following scenarios:

	Preemptive and Parallelizable	Non-Preemptive and Non-Parallelizable
Fixed Traffic	First moments are finite	First moments are finite
Heavy Traffic	Second moments are finite $\liminf_{N\to\infty} \frac{(1-\rho_N)\sqrt{N}}{\sqrt{\log\log N}} > \frac{\sqrt{2}(\sigma_m+\sigma_r)}{\sqrt{M+R}}$	Second moments are finite $\liminf_{N\to\infty} \frac{(1-\rho_N)\sqrt{N}}{\sqrt{\log\log N}} = \infty$

- Asymptotic optimality results allow multiple phases scenario (Not limited to Map and Reduce).
- Asymptotic optimality results also allow dependency among multiple phases (includes multivariate heavy tail workload).

$\rho_N \to 1$ as fast as $\frac{\sqrt{\log\log N}}{\sqrt{N}}$ $\rho_N \to 1$ slower than $\frac{\sqrt{\log\log N}}{\sqrt{N}}$

Numerical Setup

THE OHIO STATE UNIVERSITY

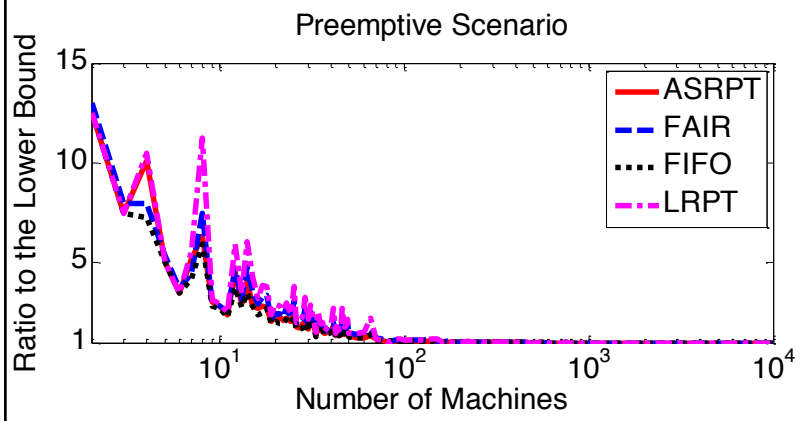
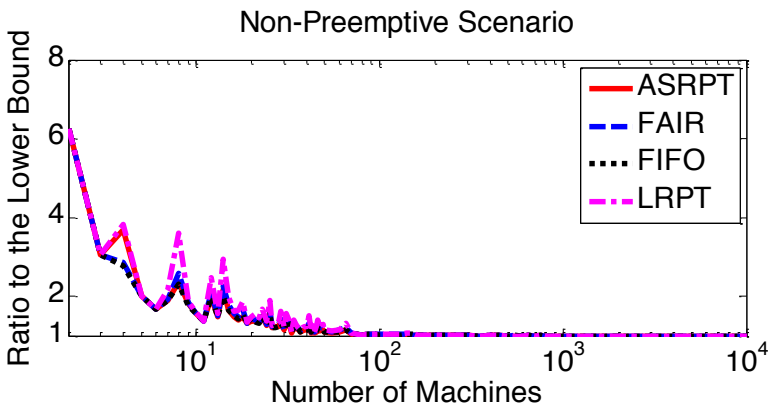
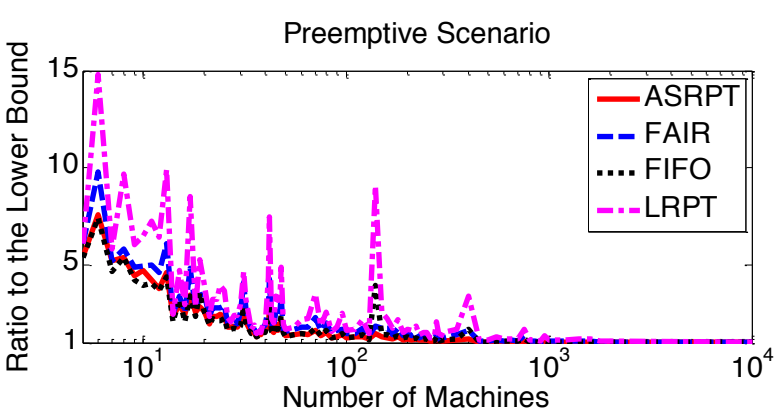
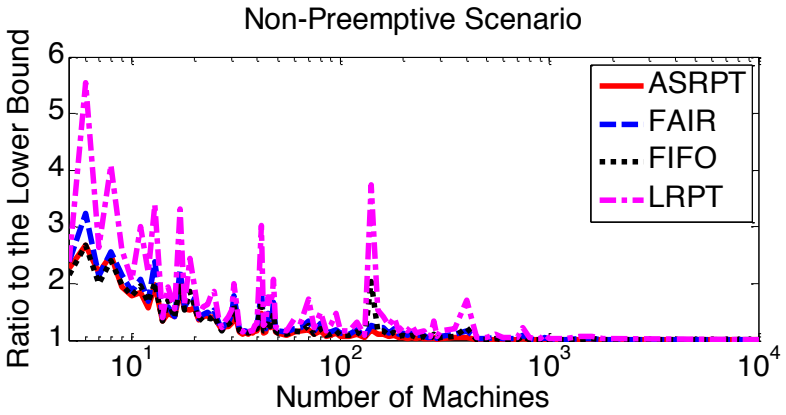
- Job arrival process: Poisson process
- Number of Map tasks and workload of Reduce jobs: Pareto Distribution

$$P(X > x) = \begin{cases} (\frac{x_m}{x})^\alpha & x \geq x_m \\ 1 & x < x_m \end{cases} .$$

- Traffic Intensity: $\rho = 0.5$
- Shape parameter: $\alpha = 3$

THE OHIO STATE UNIVERSITY

Simulation

	Preemptive and Parallelizable	Non-Preemptive and Non-Parallelizable
Fixed Traffic		
Heavy Traffic		

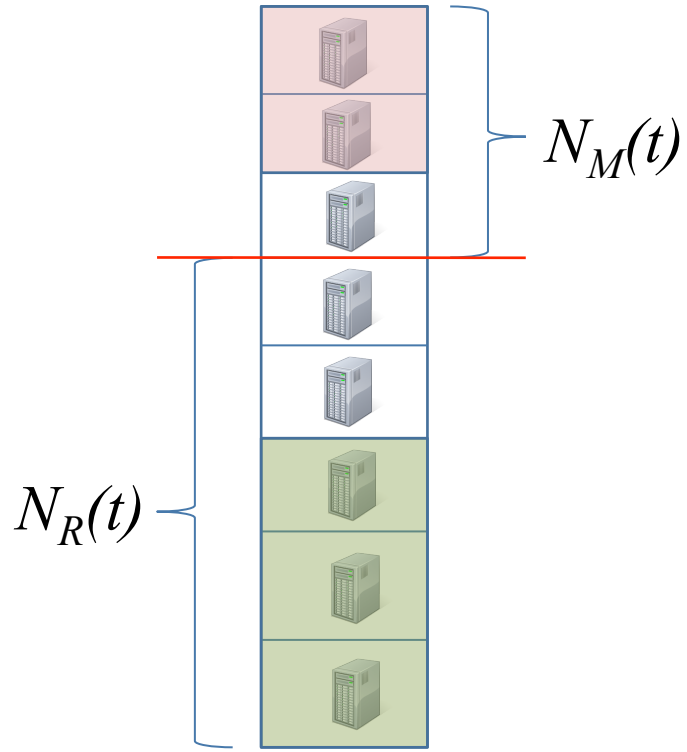
Simple but efficient

- Not all work conserving schedulers will work work well for moderate number of machines N
  - Important scenario because distributed data centers are becoming quite popular
- Question: Can we develop a simple scheduler that requires no information about the workload (statistical or real-time) and works well for a large range of N?
- Dynamic Queue-Length Based Scheduler
  - A dynamic threshold is determined at each time slot to determine the pool of machines to serve Map and Reduce jobs
  - Threshold is based on the ratio of the queue lengths (# of jobs) between Map and Reduce
    - Intuition: Tries to approximate the threshold based on the workload ratio between Map and Reduce without knowledge of the workload statistics

Dynamic Queue-Length based Scheduler

- Define $Q_M(t)$ to be the number of Map jobs and $Q_R(t)$ to be the number of Reduce jobs (new and remaining) in the system at time-slot *t*.
- Then at time t, let the pool of MAP machines be $N_M(t)$ and the pool of Reduce machines be $N_R(t)$, given as:

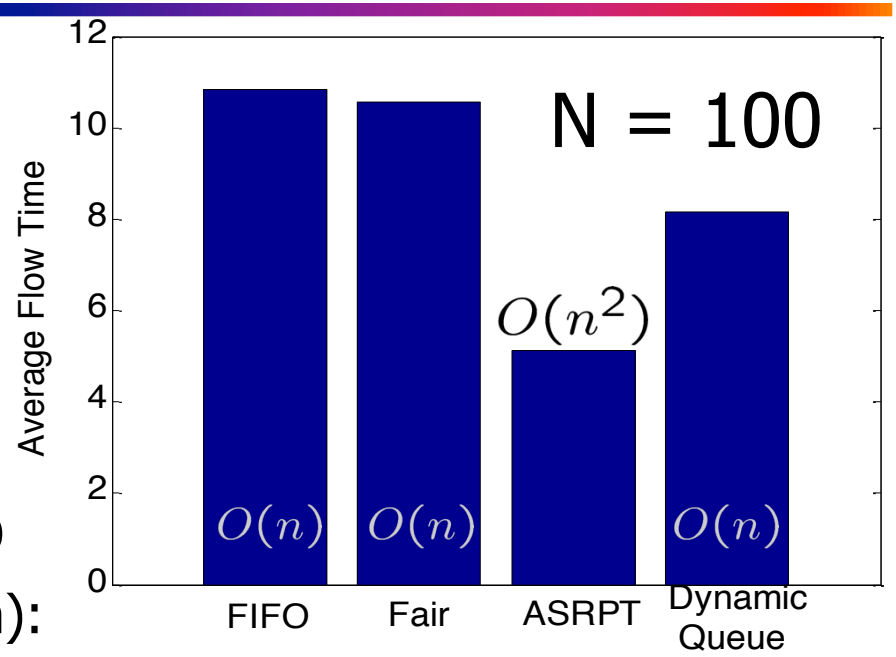
$$\begin{cases} N_M(t) = \frac{Q_M(t)}{Q_M(t)+Q_R(t)}N \\ N_R(t) = \frac{Q_R(t)}{Q_M(t)+Q_R(t)}N \end{cases}$$

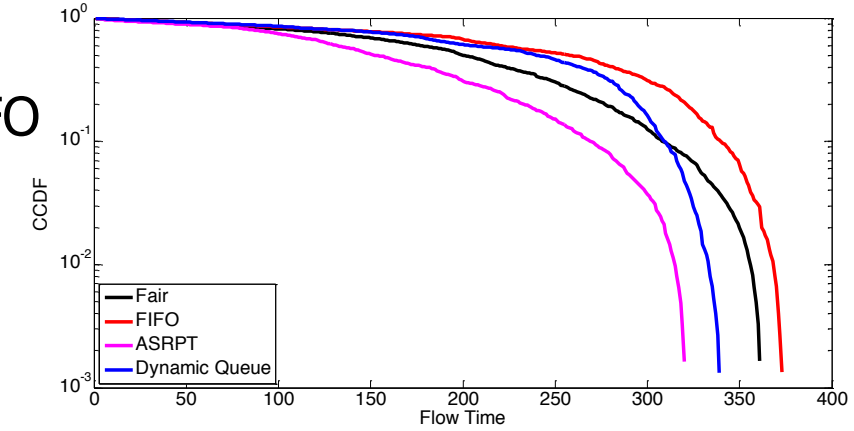


- Map jobs have higher priority in the pool of Map machines
- Reduce jobs have higher priority in the pool of Reduce machines
- Preemptive and parallelizable: If there is idle space in either phases, then cross the threshold. (Work-conserving)
- Non-preemptive and non-parallelizable: Map can cross the threshold, Reduce cannot. (Not work-conserving)

Numerical Comparion

- Schedulers
  - ASRPT
  - Fair
  - FIFO
  - Dynamic Queue
- Flow time:
  ASRPT<Dynamic Queue<Fair, FIFO
- Complexity (in terms of # of jobs n):
  ASRPT ($O(n^2)$)>Dynamic Queue($O(n)$)=Fair, FIFO ($O(n)$)
- CCDF tail:
  ASRPT<Dynamic Queue<Fair<FIFO

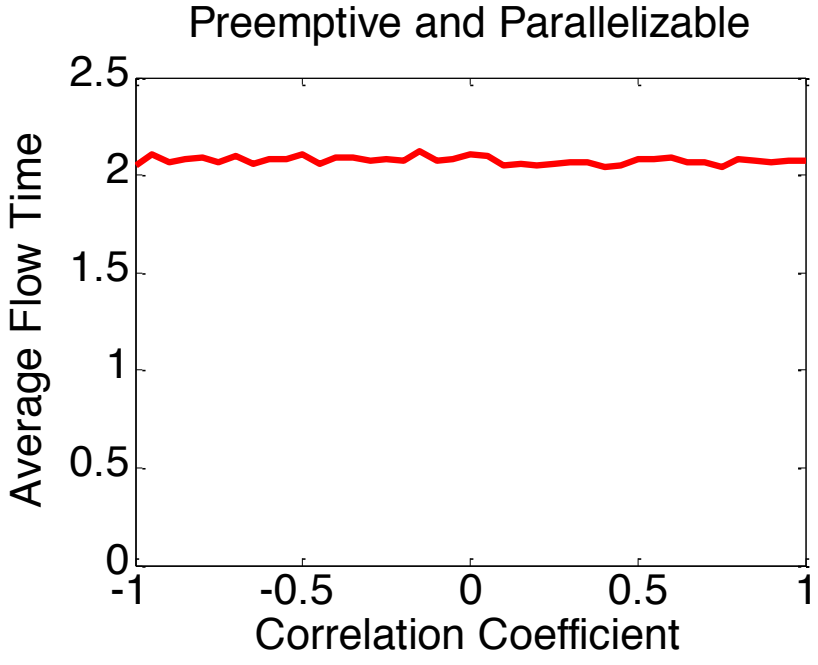


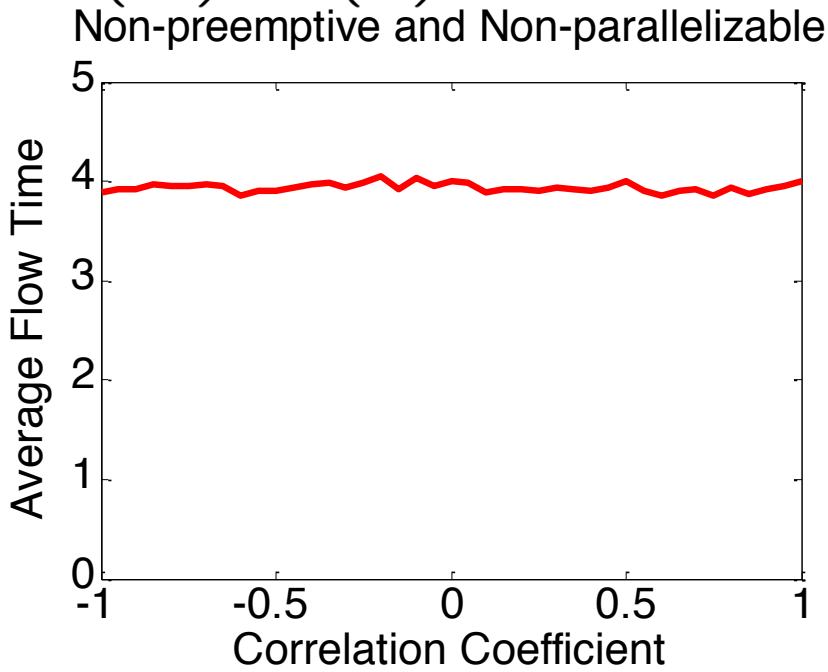


Robustness to Dependency

THE OHIO STATE UNIVERSITY

- Dynamic Queue Length Based Scheduler
- Pareto distribution: $\alpha = 3$
- Correlation Coefficient: $\frac{Cov(M,R)}{\sqrt{Var(M)Var(R)}}$



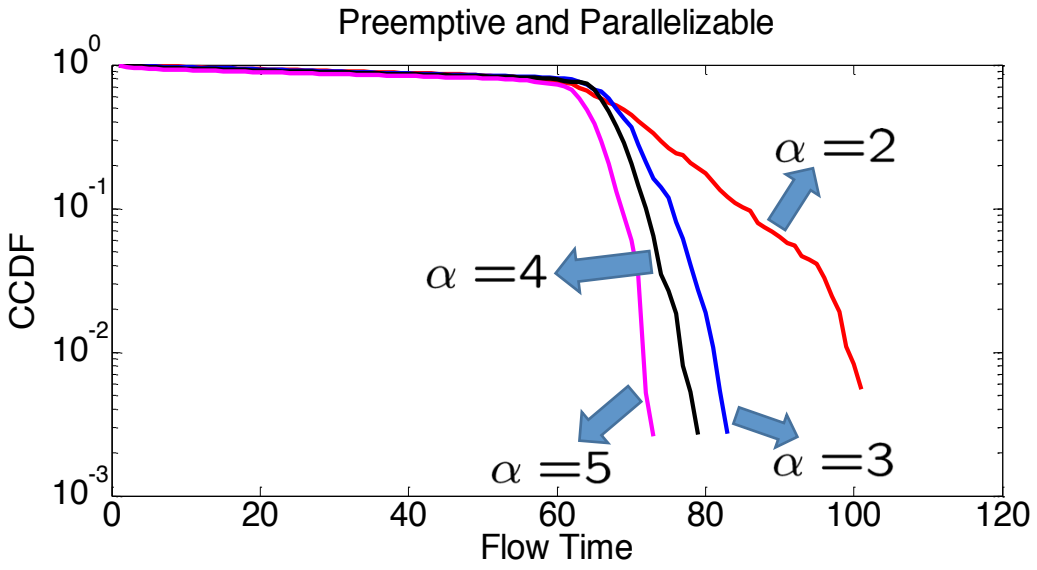


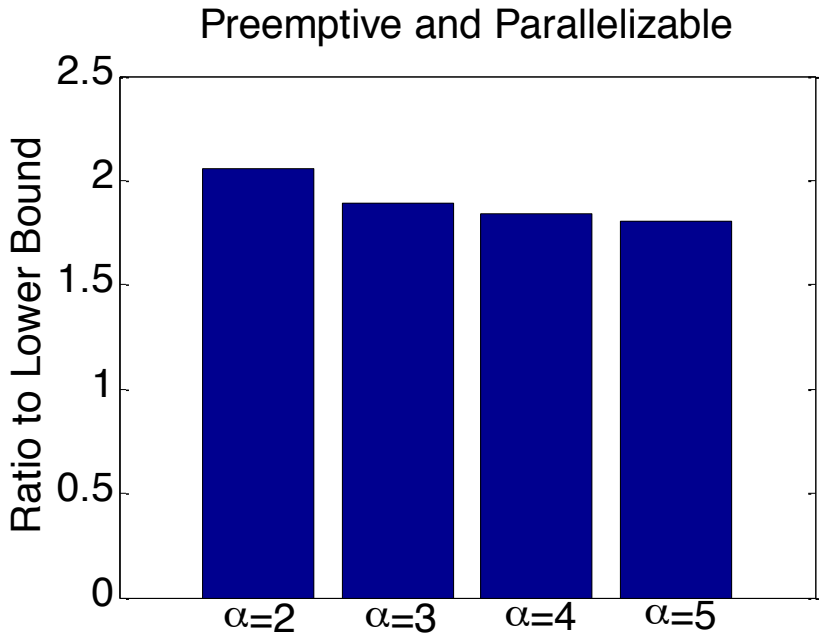
Scheduler performance appears to be invariant to the level of dependency (dynamic scheduler is robust)

Robustness to Heaviness of the Tail

THE OHIO STATE UNIVERSITY

- Dynamic Queue Length Based Scheduler
- Pareto distribution: $\alpha = 2, 3, 4, 5$



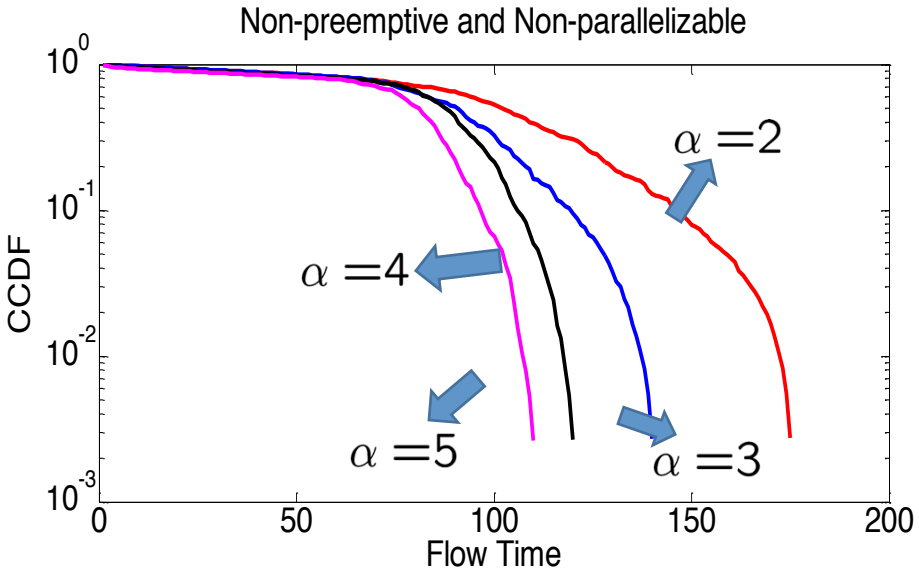


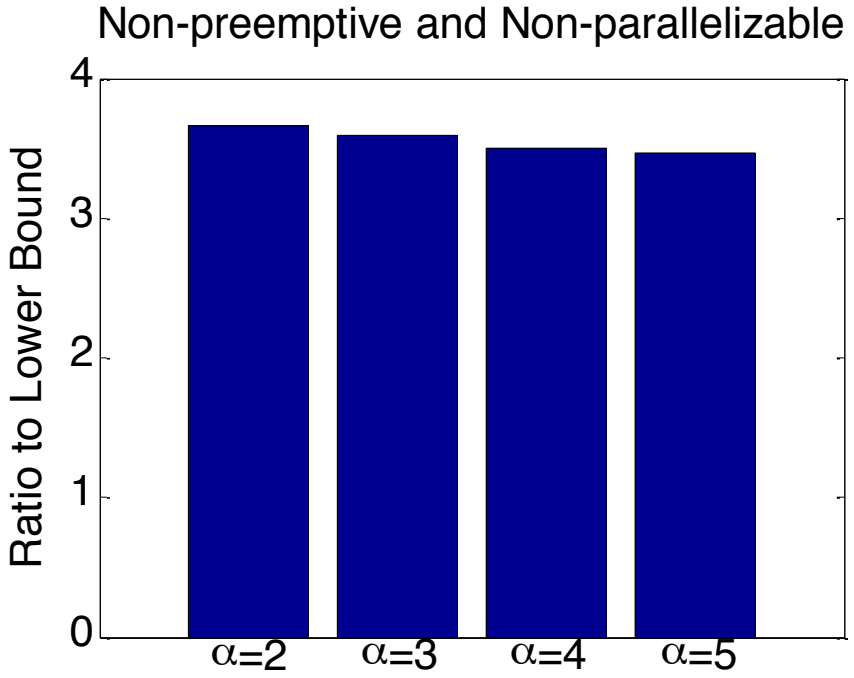
- While the delay increases with a heavier tail (workload), the relative performance (e.g., efficiency ratio) stays relatively flat.

Robustness to Heaviness of the Tail

THE OHIO STATE UNIVERSITY

- Dynamic Queue Length Based Scheduler
- Pareto distribution: $\alpha = 2, 3, 4, 5$





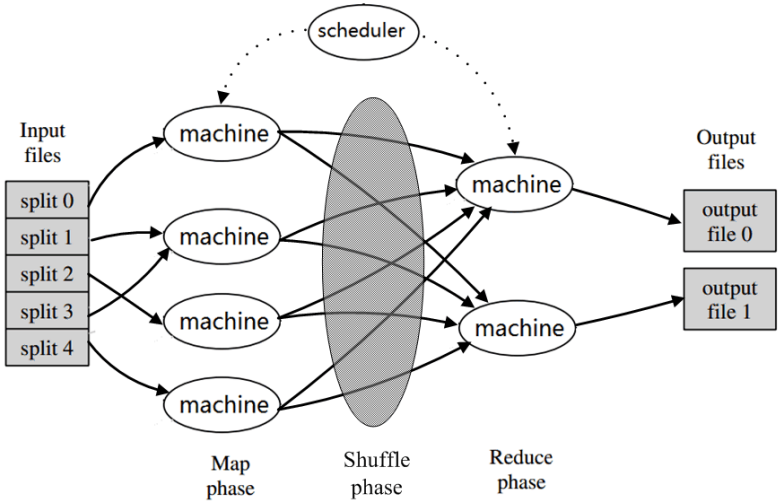
- While the delay increases with a heavier tail (workload), the relative performance (e.g., efficiency ratio) stays relatively flat.

Summary

- Flow-time minimization problem is strongly NP-hard
  - No online policy has a bounded competitive ratio (non-pre-emptive)
- All work-conserving schedulers have a constant efficiency ratio in both preemptive and non-preemptive scenarios.
  - Efficiency Ratio based analysis provides worse case (w.p.1) guarantees, but gives more design flexibility than competitive ratio
- A specific algorithm (ASRPT) can guarantee an efficiency ratio of 2 in the preemptive scenario & works well for all tested cases.
- Under the large-system limit:
  - Showed that work conserving schedulers are optimal for both fixed traffic intensity and heavy traffic scenarios
- Developed a simple dynamic threshold based policy that works well for moderate sized data centers
- Our developed algorithms appear to be robust to:
  - Varying degrees of heavy tails (varying shape parameter)
  - Varying levels of dependencies (varying the correlation coefficient)

Ongoing/Future Work

- Consider cost of data migration
  - Combine preemptive and non-preemptive scenarios
- Consider Shuffle Phase
  - Introduce more information (e.g., dependency Graph) to the scheduler
  - So that all reduce tasks don't wait until Map tasks are completed



- Find schedulers with highest rate of convergence to optimal
  - Find schedulers that can maximize the CCDF decay slope…
- Design delay based workload agnostic schedulers
  - Without detailed knowledge of Map or Reduce workload
- Consider networks of data centers
  - Scheduling also includes delays to fetch data from different servers

Collaborations/Synergistic Activities

THE OHIO STATE UNIVERSITY

- R. Srikant (UIUC)
  - Jointly supervise OSU PhD student Yousi Zheng via weekly (Thu. 11AM Eastern) Skype meetings on data center scheduling problems
  - UIUC PhD student Siva Theja Maguluri visited OSU to collaborate on autocorrelation characterization in cloud computing.
  - Visit to OSU, kickoff, and Columbia meetings, plus phone conferences to discuss mobility modeling and coupled queuing problems
  - One submitted paper, and others in progress.
- Yoora Kim (Math dept., University of Ulsan)
  - Weekly Skype meetings with OSU PhD student Irem Koprulu via weekly (Tue. 10AM Eastern) on Lévy flight analysis and explore/return model
  - Joint investigation of data-off loading problem
- Zhi-Li Zhang (University of Minnesota)
  - Kickoff and Columbia meetings + phone/email conferences to discuss human mobility modeling
  - Provided important new references on human mobility

Publications

THE OHIO STATE UNIVERSITY

1. Y. Kim, I. Koprulu and N. B. Shroff, "First Exit Time of a Levy Flight from a Bounded Region," accepted to **Advances in Applied Probability, 2014**.
2. Y. Zheng, P. Sinha, and N. B. Shroff, "A New Analytical Technique for Designing Provably Efficient MapReduce Schedulers," **IEEE INFOCOM'13**, April 2013, Turin, Italy.
3. H. Cai, I. Koprulu, and N. B. Shroff "Exploiting Double Opportunities for Deadline Based Content Propagation in Wireless Networks," **IEEE INFOCOM'13,** April 2013, Turin, Italy (extended version submitted for journal publication).
4. Y. Kim, K. Lee, and N. B. Shroff, "An Analytical Framework to Characterize the Efficiency and Delay in a Mobile Data Offloading System," **ACM Mobihoc'14**, Philadelphia, PA, August 2014.
5. S. Buccapatnam, A. Eryilmaz, N. B. Shroff, "Stochastic Bandits with Side Observations on Networks," **ACM SIGMETRICS'14,** June 2014, Austin, Texas.
6. Y. Sun, Z. Zhang, E. Koksal, K-H. Lee, N. B. Shroff "Provably Delay Efficient Data Retrieving in Storage Clouds," **submitted** for publication, July 2014.
7. Y. Zheng, N. B. Shroff, **R. Srikant,** and P. Sinha, "Exploiting Large System Dynamics for Designing Simple Data Center Schedulers," **submitted** for publication, July 2014.

Thank You!