Scheduling Stochastic Jobs on HPC Platforms (and Beyond)

Speaker: Hongyang Sun (Vanderbilt University, INRIA)

Joint work with: Guillaume Aupy (INRIA, Univ. Bordeaux), Ana Gainaru (Vanderbilt University), Valentin Honoré (INRIA, Univ Bordeaux), Padma Raghavan (Vanderbilt University), Yves Robert (INRIA, ENS de Lyon, UTK)

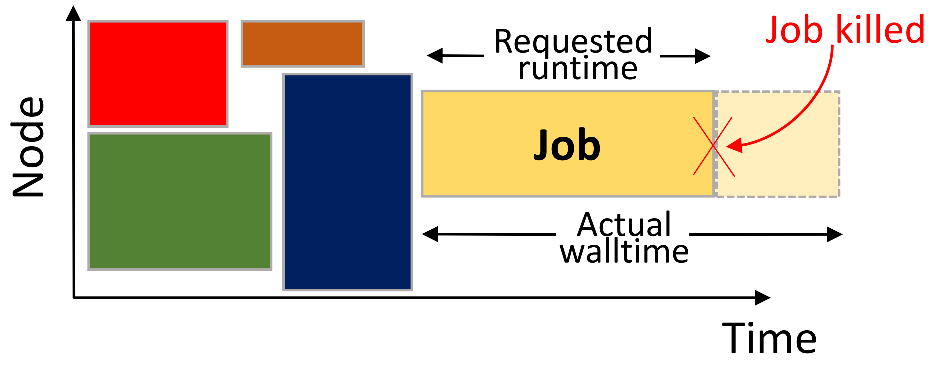
9th JLECS Workshop, Knoxville, TN, USA

April 15, 2019

HPC Batch Scheduler

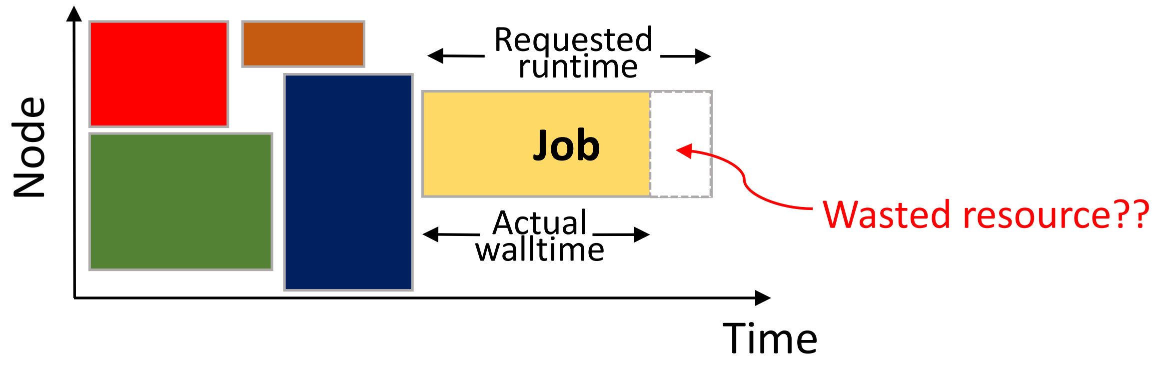
- **<u>Reservation-Based</u>:**
  - ❑ Relies on (reasonably) accurate runtime estimation from the user/application
  - ❑ Intended for HPC jobs with (relatively) *deterministic* and *predictive* behavior

<u>Resource under-estimation</u>



- ➢ Job killed; need to resubmit; prolonged completion time
- ➢ Waste of system resources

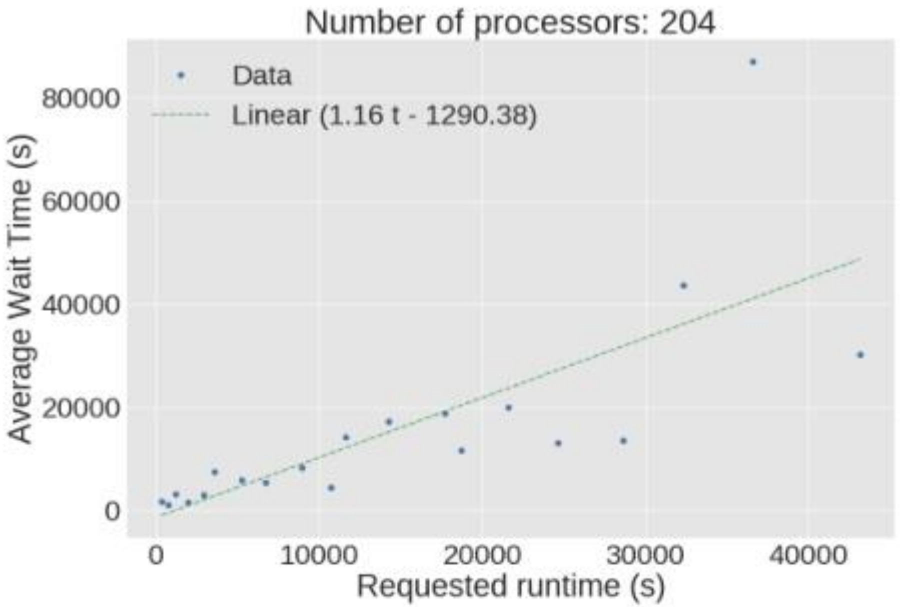
<u>Resource over-estimation</u>



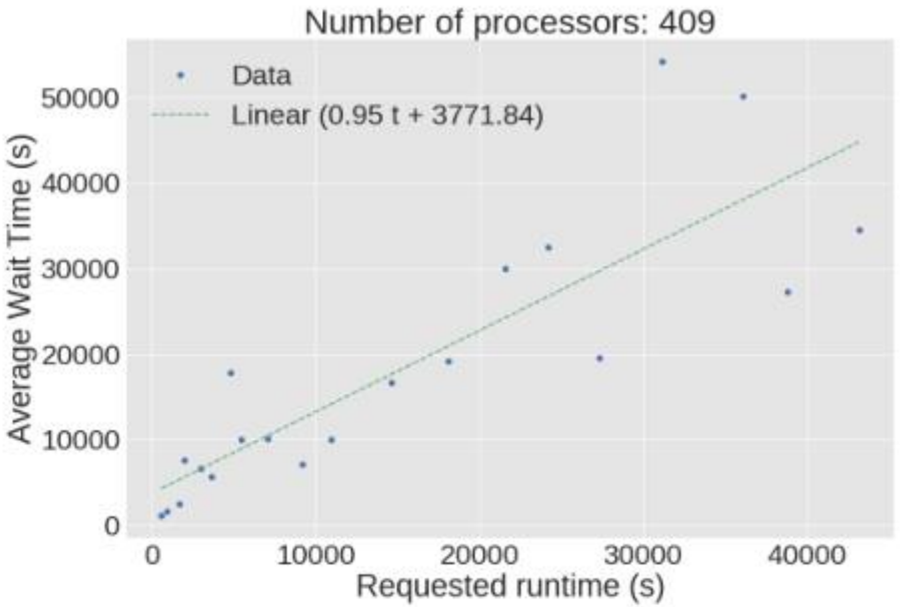
- ➢ Job completed early; but may have waited longer in queue than needed
- ➢ May waste system resources (if no backfilling possible)

Computing in HPC

Execution Time = Wait Time + Runtime



(a) Jobs that requested 204 procs.

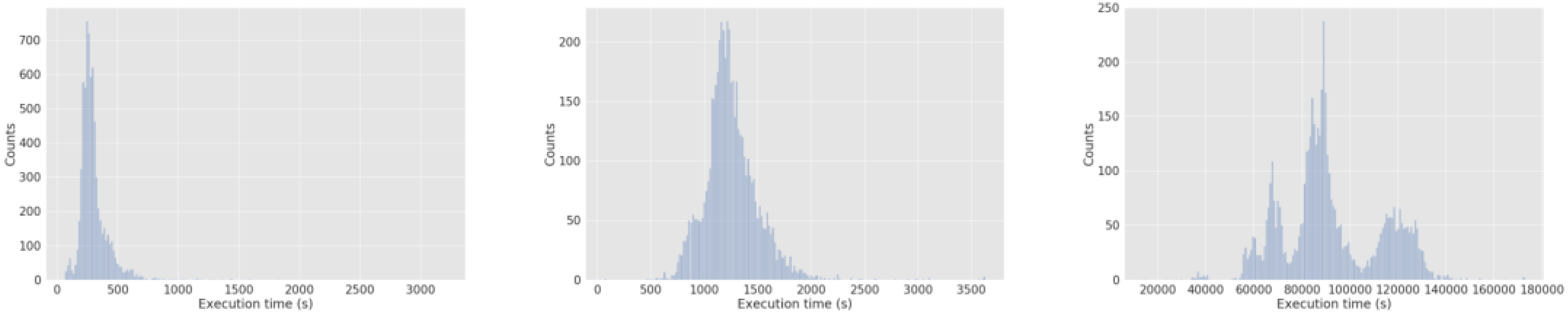


(b) Jobs that requested 409 procs.

Figure: Average wait times of jobs run on Intrepid (2009) as a function of requested runtime (data: Parallel Workload Archive).

Stochastic Jobs

- Many scientific applications are ***stochastic*** and ***unpredictable***
  - ❑ Execution time is *input-dependent* (stochastic)
  - ❑ Unpredictable even for *same input-size* (quality matters)
  - ❑ *Large variations* (order of magnitude difference)
  - ❑ Common in *many domains* (e.g. neuroscience, adaptive mesh refinement)

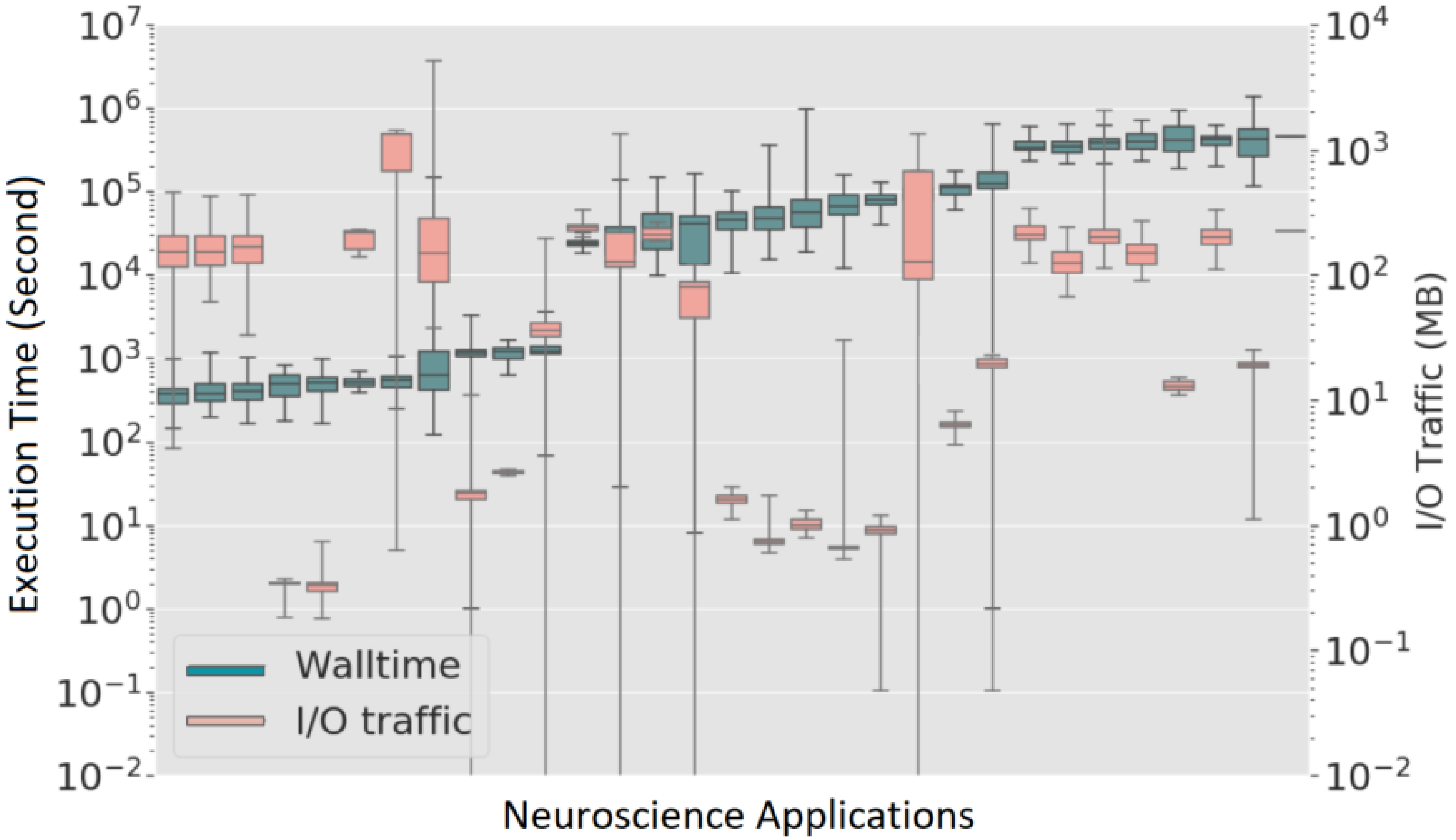


(a) fMRIQA (b) VBMQA (c) dtiQA

Figure: Traces [2013-2016] of neuroscience apps (Vanderbilt's medical imaging database).

Neuroscience Applications

Range of execution times and I/O traffics for 31 representative neuroscience applications



Coping with Stochastic Jobs

- **Scheduling Options:**
  - ❑ ***System-level solution***:
    - Abandon reservation-based batch scheduling
    - Use online (on-the-fly) scheduling ➔ **not practical**
  - ❑ ***Application-level solution***:
    - Develop optimized code to reduce stochasticity
    - Better resource estimation (e.g., using ML methods) ➔ **difficult**

Coping with Stochastic Jobs

- **<u>Scheduling Options:</u>**
  - ❑ ***System-level solution***:
    - - Abandon reservation-based batch scheduling
    - - Use online (on-the-fly) scheduling ➔ **not practical**
  - ❑ ***Application-level solution***:
    - - Develop optimized code to reduce stochasticity
    - - Better resource estimation (e.g., using ML methods) ➔ **difficult**
  - ❑ ***Our approach***:
    - - Optimization of expected job execution times
    - - **Non-disruptive** to existing HPC scheduling model and application development process

Computing in the Cloud

- <u>**Several Pricing Models** *(e.g., using Amazon AWS)*</u>
  - ❑ **On-Demand (OD) = pay-what-you-use**:

    "you pay for compute capacity by the hour or second depending on which instances you run"
  - ❑ **Reserved-Instances (RI) = Pay-what-you-reserve**:

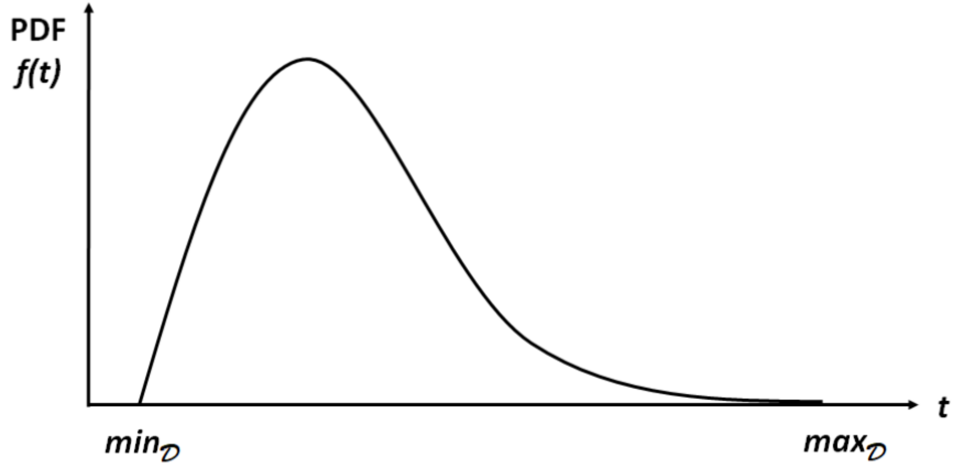
    "provide you with a significant discount (up to 75%) compared to On-Demand pricing"

Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
No Upfront	$0.00	$8.03	$0.011	57%	
Partial Upfront	$134.00	$3.72	$0.01	60%	$0.0255
All Upfront	$252.00	$0.00	$0.01	62%	

Data extracted from AWS website

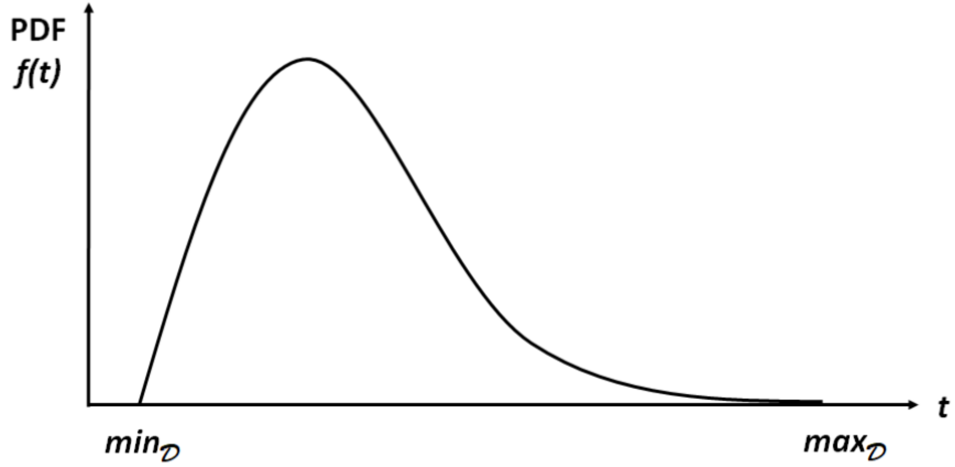
Models

- **<u>Job Model</u>:** Execution time modeled by a random variable X that follows:
  - ❑ Known probability distribution *D*
  - ❑ PDF = *f(t)* and CDF = *F(t)*
  - ❑ Positive support: $X \in [\min_D, \max_D]$



Models

- **<u>Job Model</u>:** Execution time modeled by a random variable X that follows:
  - ❑ Known probability distribution *D*
  - ❑ PDF = *f(t)* and CDF = *F(t)*
  - ❑ Positive support: $X \in [\min_D, \max_D]$



- **<u>Cost Model:</u>** If reserve $\boldsymbol{t_1}$ time and actual execution is $\boldsymbol{t}$ time:

$$\boxed{Cost = \underbrace{\alpha t_1}_{\text{Reservation cost}} + \underbrace{\beta \min(t_1, t)}_{\text{Usage cost}} + \underbrace{\gamma}_{\text{Setup cost}}}$$

  - ❑ If $\boldsymbol{t_1} \geq \boldsymbol{t}$, then reservation is enough and job succeeds
  - ❑ If $\boldsymbol{t_1} < \boldsymbol{t}$, then job is killed; a new reservation ($\boldsymbol{t_2} > \boldsymbol{t_1}$) is needed

Optimization Objective

- The objective is to compute a sequence of ***increasing reservations***:

$$S = (t_1, t_2, \ldots, t_i, t_{i+1}, \ldots)$$

that minimizes the ***total expected cost***:

$$\boxed{\mathbb{E}(S) = \beta \cdot \mathbb{E}[X] + \sum_{i=0}^{\infty} (\alpha t_{i+1} + \beta t_i + \gamma)\, \mathbb{P}(X \geq t_i)}$$

Expected total usage cost

Extra cost incurred for each failed reservation

Solution 1: Characterizing Optimal Sequence

- **Existence**: *optimal sequence (with finite expected cost) exists for distributions with bounded mean and variance*

Solution 1: Characterizing Optimal Sequence

- **Existence**: *optimal sequence (with finite expected cost) exists for distributions with bounded mean and variance*
- **Property**: *optimal sequence satisfies the following recursive relationship for smooth distributions:*

$$\boxed{t_i^o = \frac{1 - F(t_{i-2}^o)}{f(t_{i-1}^o)} + \frac{\beta}{\alpha}\left(\frac{1 - F(t_{i-1}^o)}{f(t_{i-1}^o)} - t_{i-1}^o\right) - \frac{\gamma}{\alpha}}$$

  - ❑ Compute t_i based on t_{i-1} and t_{i-2} (*as in Fibonacci numbers*)
  - ❑ By default $t_0 = 0$, it remains to compute t_1
  - ❑ Bounded search range: $t_1^o \in [\min_D, O(\text{mean} + \text{var})]$
  - ❑ Complexity of computing optimal t_1^o is unclear (rational solution)

Solution 1: Characterizing Optimal Sequence

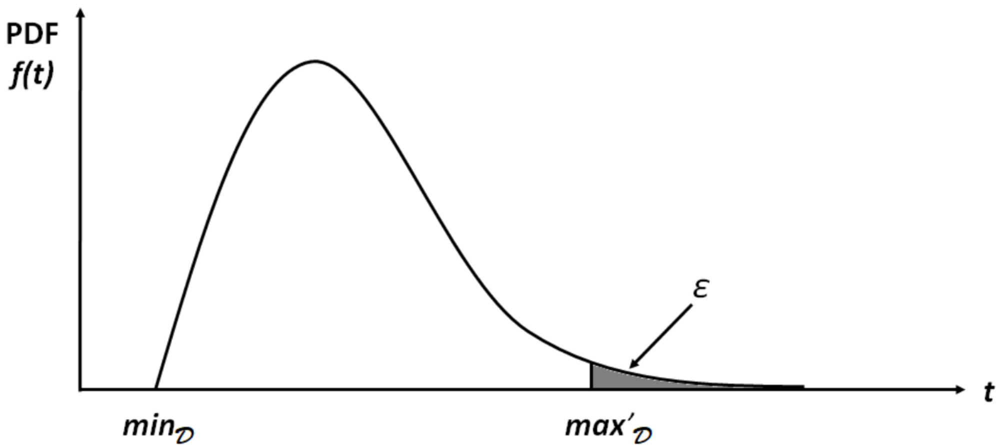
- **Existence**: *optimal sequence (with finite expected cost) exists for distributions with bounded mean and variance*
- **Property**: *optimal sequence satisfies the following recursive relationship for smooth distributions:*

$$\boxed{t_i^o = \frac{1 - F(t_{i-2}^o)}{f(t_{i-1}^o)} + \frac{\beta}{\alpha}\left(\frac{1 - F(t_{i-1}^o)}{f(t_{i-1}^o)} - t_{i-1}^o\right) - \frac{\gamma}{\alpha}}$$

  - ❑ Compute t_i based on t_{i-1} and t_{i-2} (*as in Fibonacci numbers*)
  - ❑ By default $t_0 = 0$, it remains to compute t_1
  - ❑ Bounded search range: $t_1^o \in [\min_D, O(\text{mean} + \text{var})]$
  - ❑ Complexity of computing optimal t_1^o is unclear (rational solution)
- **Heuristic (Brute-Force)**: *Numerical search of optimal t_1^o in the range*

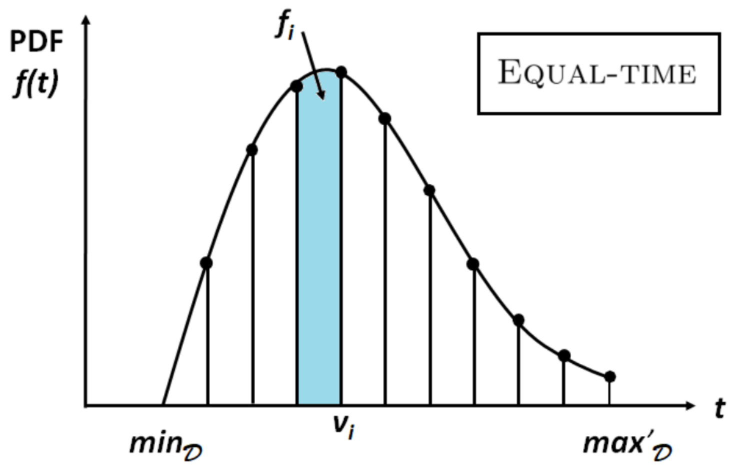
Solution 2: Approximating via Discretization

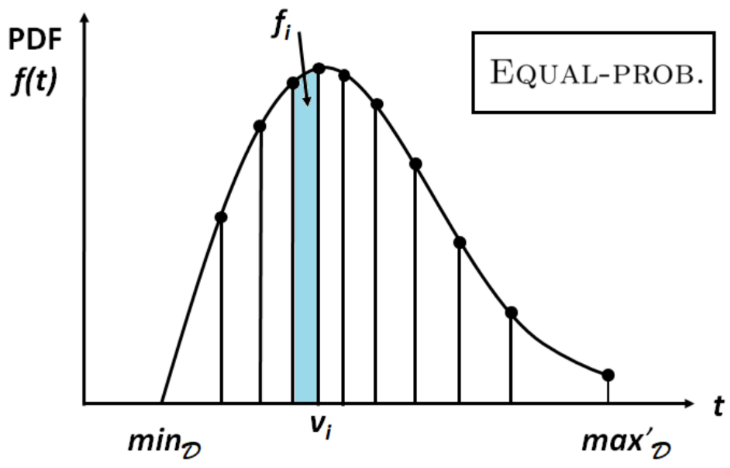
- **Discrete Transformation**: *truncate and discretize continuous distribution*



Solution 2: Approximating via Discretization

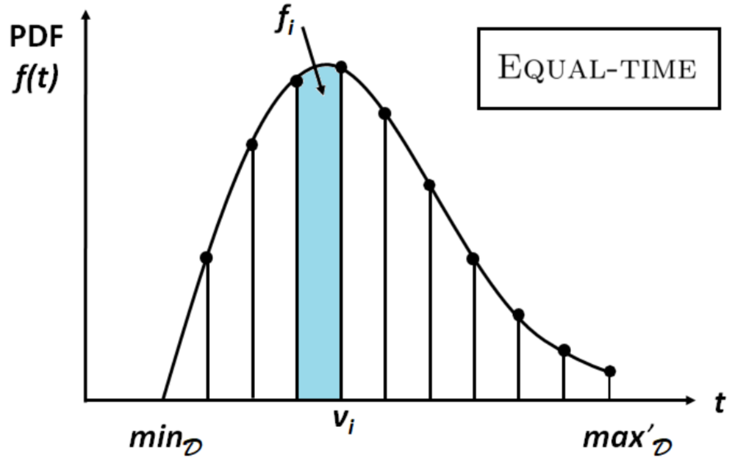
- **Discrete Transformation**: *truncate and discretize continuous distribution*

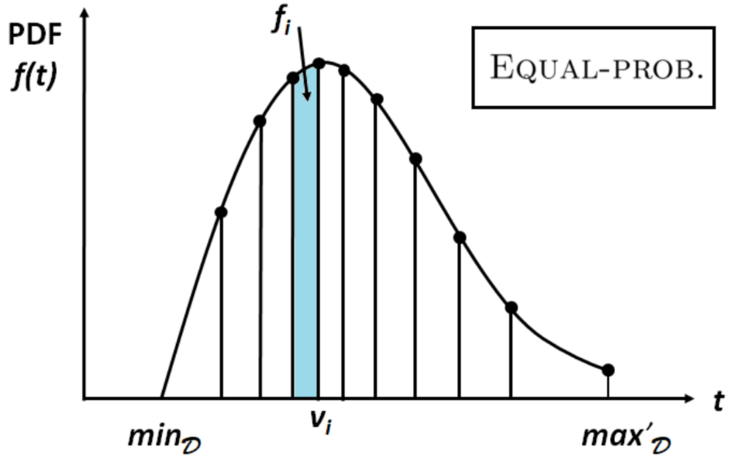




Solution 2: Approximating via Discretization

- **Discrete Transformation**: *truncate and discretize continuous distribution*





- **Dynamic Programming**: *for discrete distribution* $X \sim (v_i, f_i)_{i=1..n}$

$$\boxed{\mathbb{E}_i^* = \min_{i \le j \le n} \left(\alpha v_j + \gamma + \sum_{k=i}^{j} f'_k \cdot \beta v_k + \left(\sum_{k=j+1}^{n} f'_k \right) \left(\beta v_j + \mathbb{E}_{j+1}^* \right) \right)}$$

Cost of successful reservation

Cost of failure ➔ sub-problem

- ❑ Initialization:

$$\mathbb{E}_n^* = \alpha v_n + \beta v_n + \gamma$$

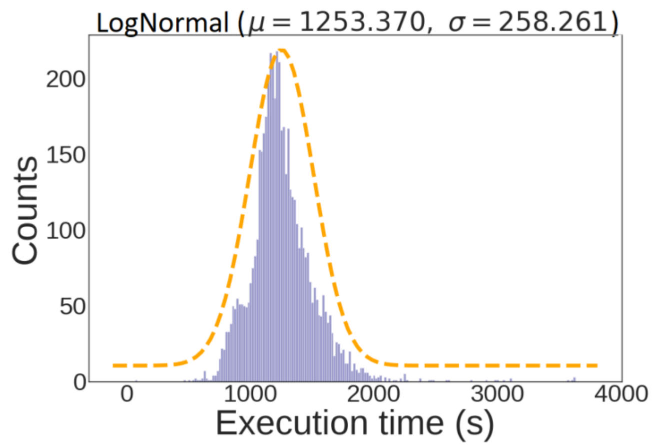
- ❑ Complexity: $O(n^2)$

Performance (Common Prob. Distributions)

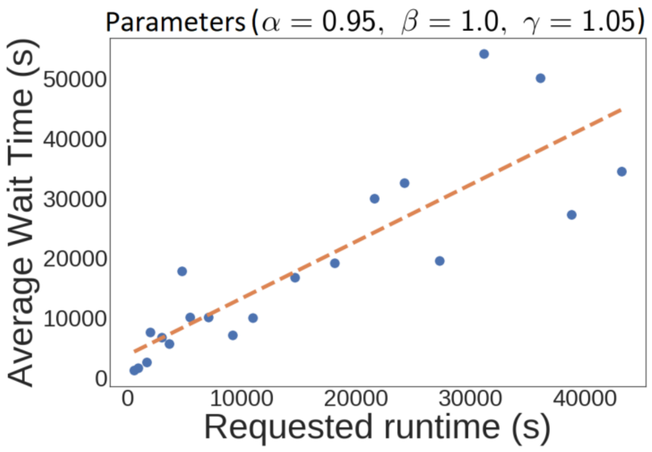
Distribution	BF(t_1)	DP(ET)	DP(EP)	MEAN-BY-MEAN	MEAN-STDEV	MEAN-DOUB.	MED-BY-MED
Exponential	2.15	2.31 (1.07)	2.36 (1.10)	2.36 (1.10)	2.39 (1.11)	2.42 (1.13)	2.83 (1.32)
Weibull	2.12	2.40 (1.13)	2.22 (1.05)	2.76 (1.30)	3.58 (1.69)	3.03 (1.43)	3.05 (1.44)
Gamma	2.02	2.20 (1.09)	2.13 (1.05)	2.26 (1.12)	2.18 (1.08)	2.24 (1.11)	2.51 (1.24)
Lognormal	1.85	1.87 (1.01)	1.93 (1.04)	2.19 (1.19)	2.09 (1.13)	1.95 (1.06)	2.30 (1.24)
TruncatedNormal	1.36	1.38 (1.02)	1.36 (1.00)	1.98 (1.46)	1.83 (1.35)	1.98 (1.46)	2.16 (1.60)
Pareto	1.62	1.71 (1.05)	1.66 (1.03)	1.82 (1.12)	2.18 (1.34)	1.75 (1.08)	2.26 (1.39)
Uniform	1.33	1.33 (1.00)	1.33 (1.00)	2.21 (1.66)	1.90 (1.43)	1.67 (1.26)	2.21 (1.66)
Beta	1.75	1.79 (1.02)	1.80 (1.02)	2.02 (1.15)	2.11 (1.20)	1.98 (1.13)	2.45 (1.40)
BoundedPareto	1.80	2.00 (1.11)	1.91 (1.06)	1.84 (1.02)	2.09 (1.16)	1.83 (1.01)	2.81 (1.56)

- Brute-Force (t_1) heuristic has best performance (around 2x of offline optimal)
- Discretization-based heuristics have close performance, much better than other naïve heuristics

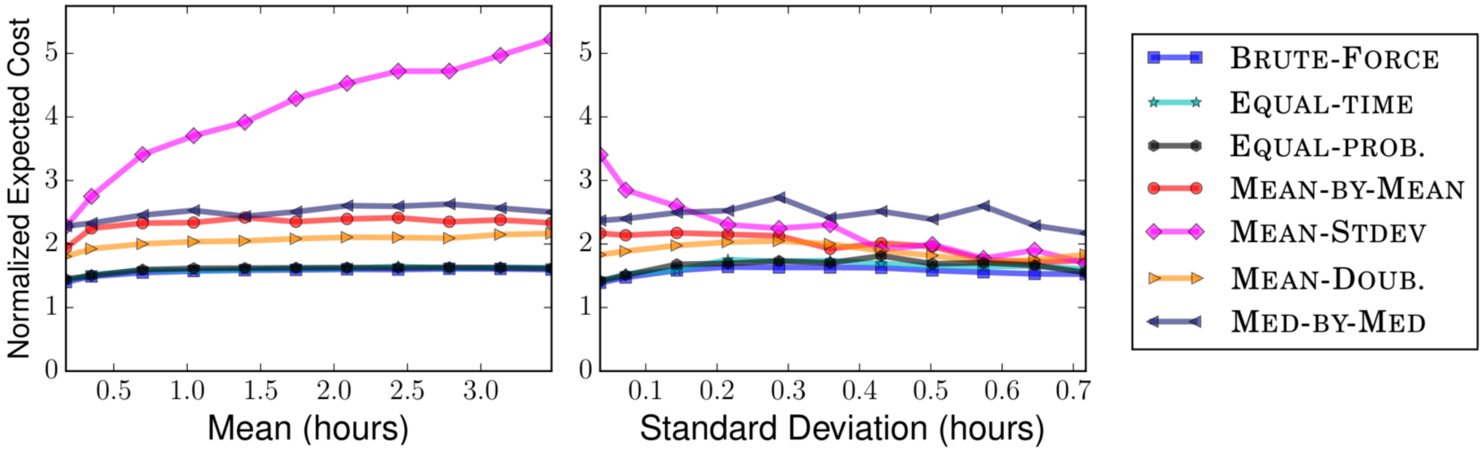
Performance (Realistic Workloads)



(a) Fitted **LogNormal** execution-time distribution for the VBMQA jobs



(b) Fitted **affine** waiting time function based on logs from the Intrepid data



(c) Performance of all heuristics with impact of varying mean and standard deviation

Future Work

- **From User's Perspective (Single Job):**
  - ❑ How to request runtime along with other resources (#nodes, memory)?
  - ❑ Is checkpointing at the end of some/all reservations useful?

    <u>Related to HPC fault tolerance</u>: Trade-off between time wasted due to checkpointing and time saved for not having to start from scratch

Future Work

- **From User's Perspective (Single Job):**
  - ❑ How to request runtime along with other resources (#nodes, memory)?
  - ❑ Is checkpointing at the end of some/all reservations useful?

    <u>Related to HPC fault tolerance</u>: Trade-off between time wasted due to checkpointing and time saved for not having to start from scratch

- **From System's Perspective (Set of Jobs):**
  - ❑ Are reservation-based schedulers still suitable for stochastic workloads?
  - ❑ How should scheduling and backfilling be performed (under uncertainty)?
  - ❑ Is it time to consider new scheduling paradigms (e.g., online, hybrid)?

    <u>Preliminary results:</u> on-the-fly scheduling better for both system-level performance (utilization) and user-level performance (average response time) for single-node stochastic jobs; work-in-progress for multi-node jobs.

Thank you!

Hongyang Sun

hongyang.sun@vanderbilt.edu

References

[1] G. Aupy, A. Gainaru, V. Honore, P. Raghavan, Y. Robert, H. Sun. (authors in alphabetical order) *Reservation Strategies for Stochastic Jobs*. IPDPS, 2019.
[2] A. Gainaru, H. Sun, G. Aupy, Y. Huo, B. Landman, P. Raghavan. *On-the-fly Scheduling vs. Reservation-based Scheduling for Unpredictable Workflows*. IJHPCA, 2019.