2023 International Conference on Advanced Computing and Analytics (ACOMPA)

ACOMPA 2023

Deviation Backfilling: A Robust Backfilling Scheme for Improving the Efficiency of Job Scheduling on High Performance Computing Systems

Thanh Hoang Le Hai , Khang Nguyen Duy,
Thin Nguyen Manh, Danh Mai Hoang and Nam Thoai

High Performance Computing Laboratory, Faculty of Computer Science & Engineering
Advanced Institute of Interdisciplinary Science and Technology
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City

Overview of HPC Systems

- Top500 (November 2023)

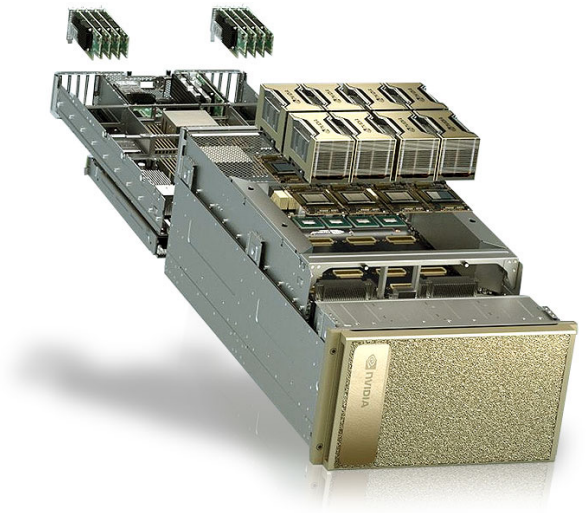
Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	4,742,808	585.34	1,059.33	24,687
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States	1,123,200	561.20	846.84	
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107

Outline

- **Introduction**
  - Overview of HPC systems
  - Scheduling on HPC Systems
  - Research Goals
- **Related Works**
- **Deviation Backfilling**
  - Deviation Delay
  - Job Runtime Estimation
- **Evaluation & Discussion**
- **Conclusion and Future Work**

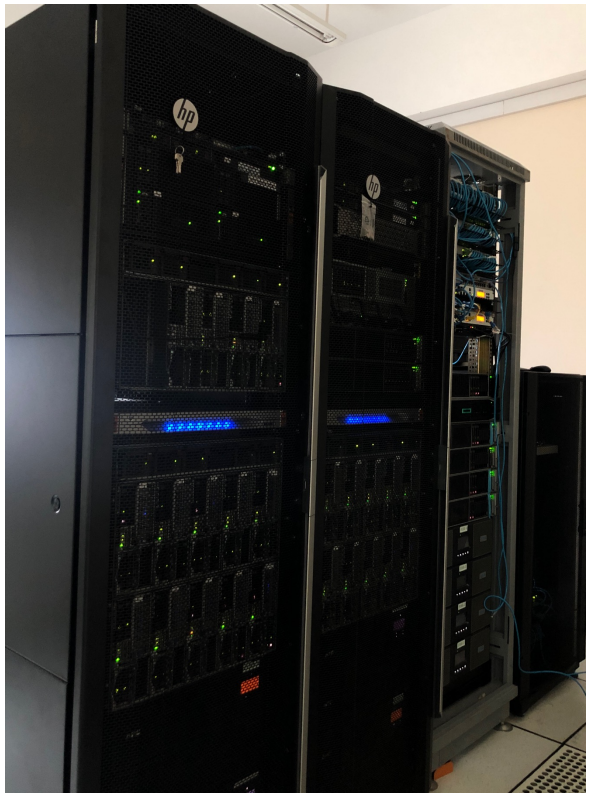
Overview of HPC Systems

- High-Performance Computing (HPC) involves using advanced hardware and parallel processing to solve complex problems at extremely high speeds.



Overview of HPC Systems

- The SuperNode-XP at HPCC, HCMUT



HPCC
HCMUT, VNU - HCM

SUPER NODE-XP

Overview of HPC Systems

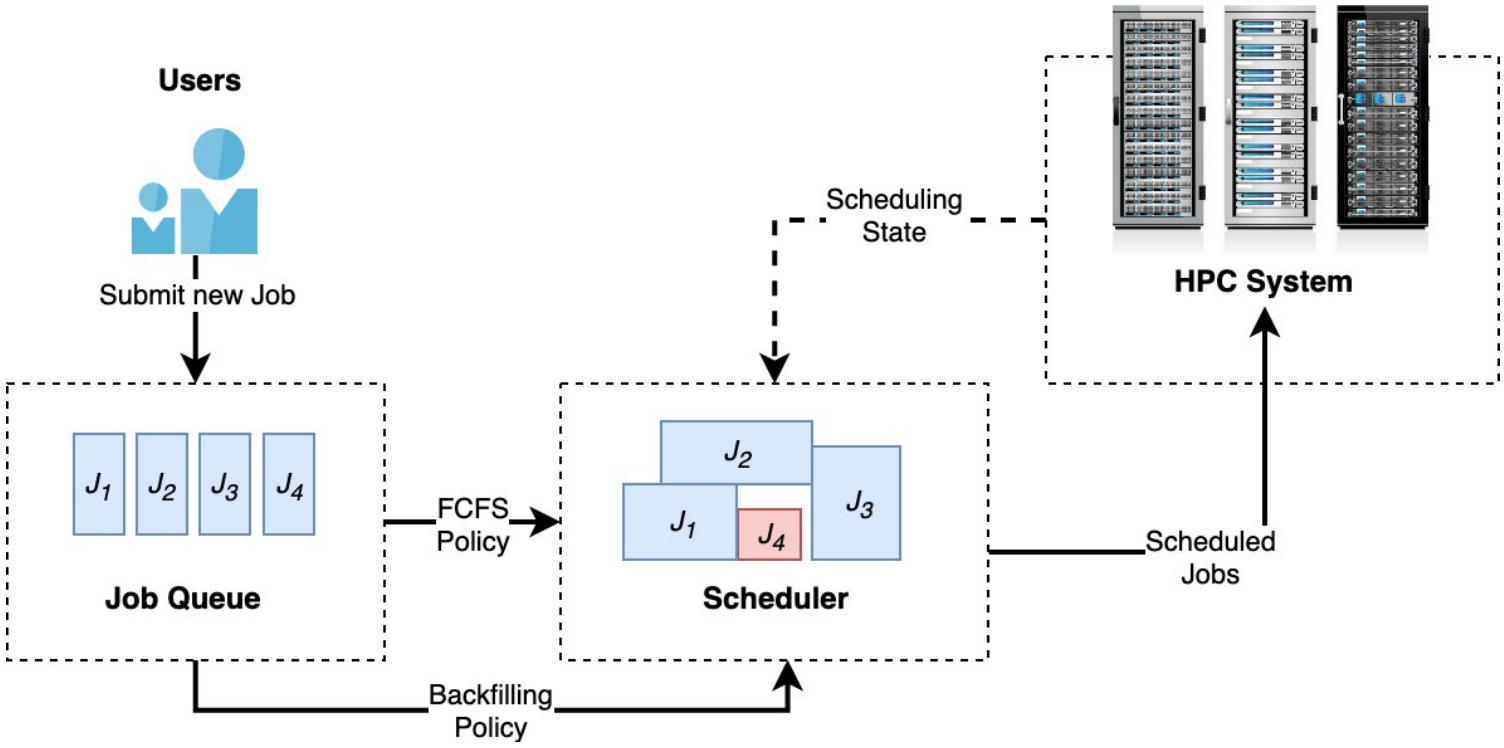
- HPCC Research Team, HCMUT

HPCC
HCMUT, VNU - HCM

www.hpcc.hcmut.edu.vn
hpcc@hcmut.edu.vn

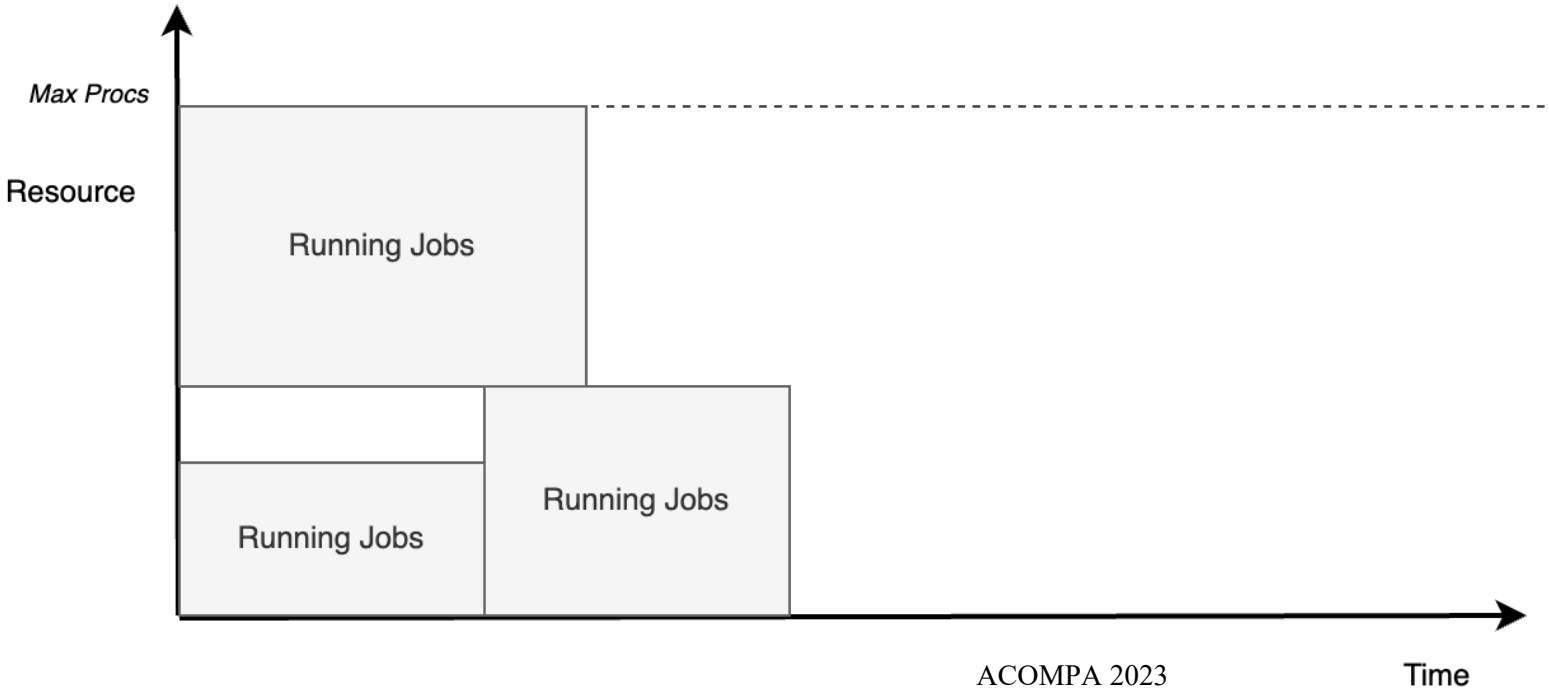
Scheduling on HPC Systems

- **Job Schedulers** help ensure fair access to computing resources while maintaining optimal system utilization.



Scheduling on HPC Systems

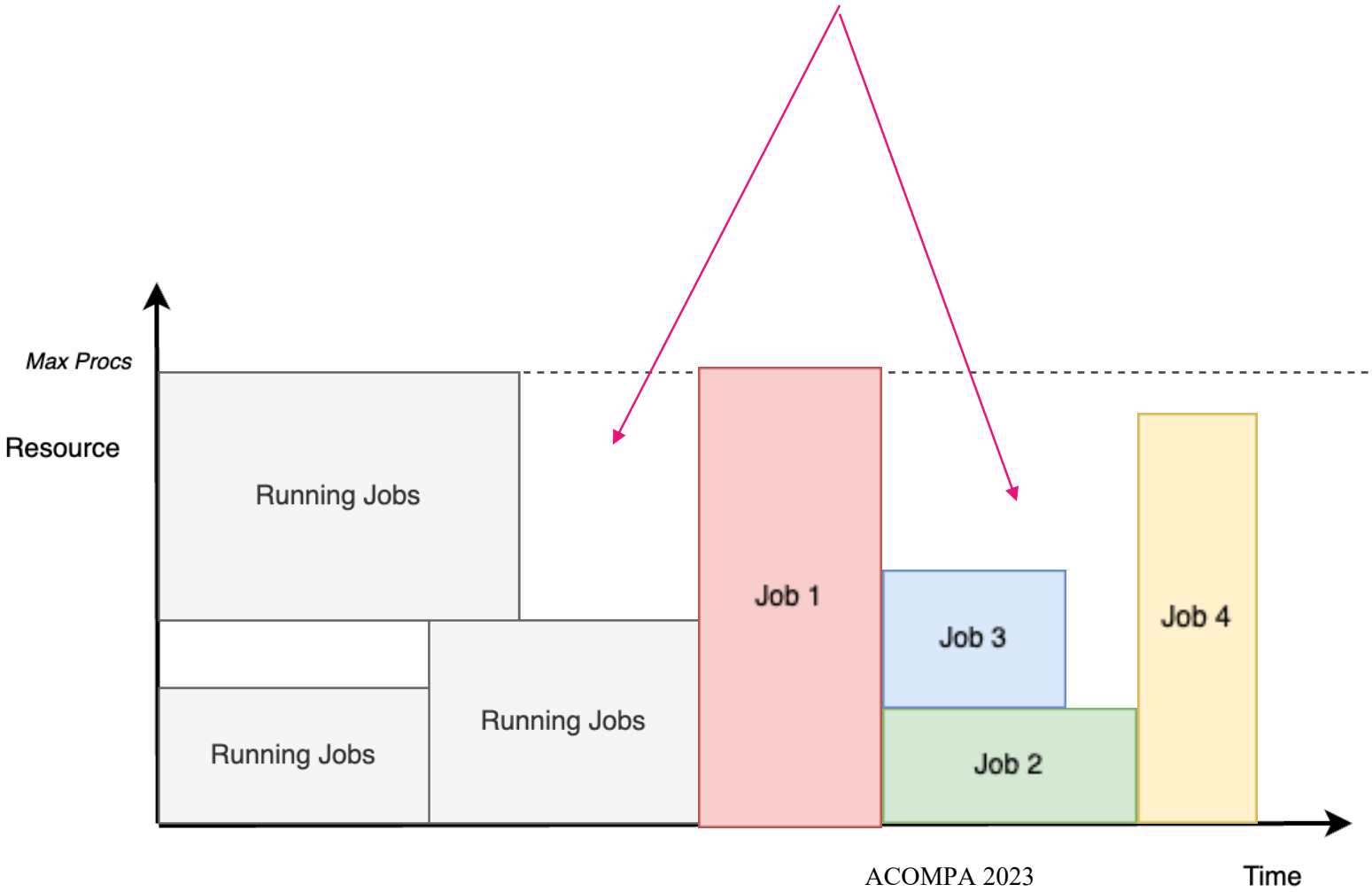
FCFS (First-Come, First-Served) is popularly used in High-Performance Computing (HPC) systems because of its simplicity and fairness in resource allocation



Scheduling on HPC Systems

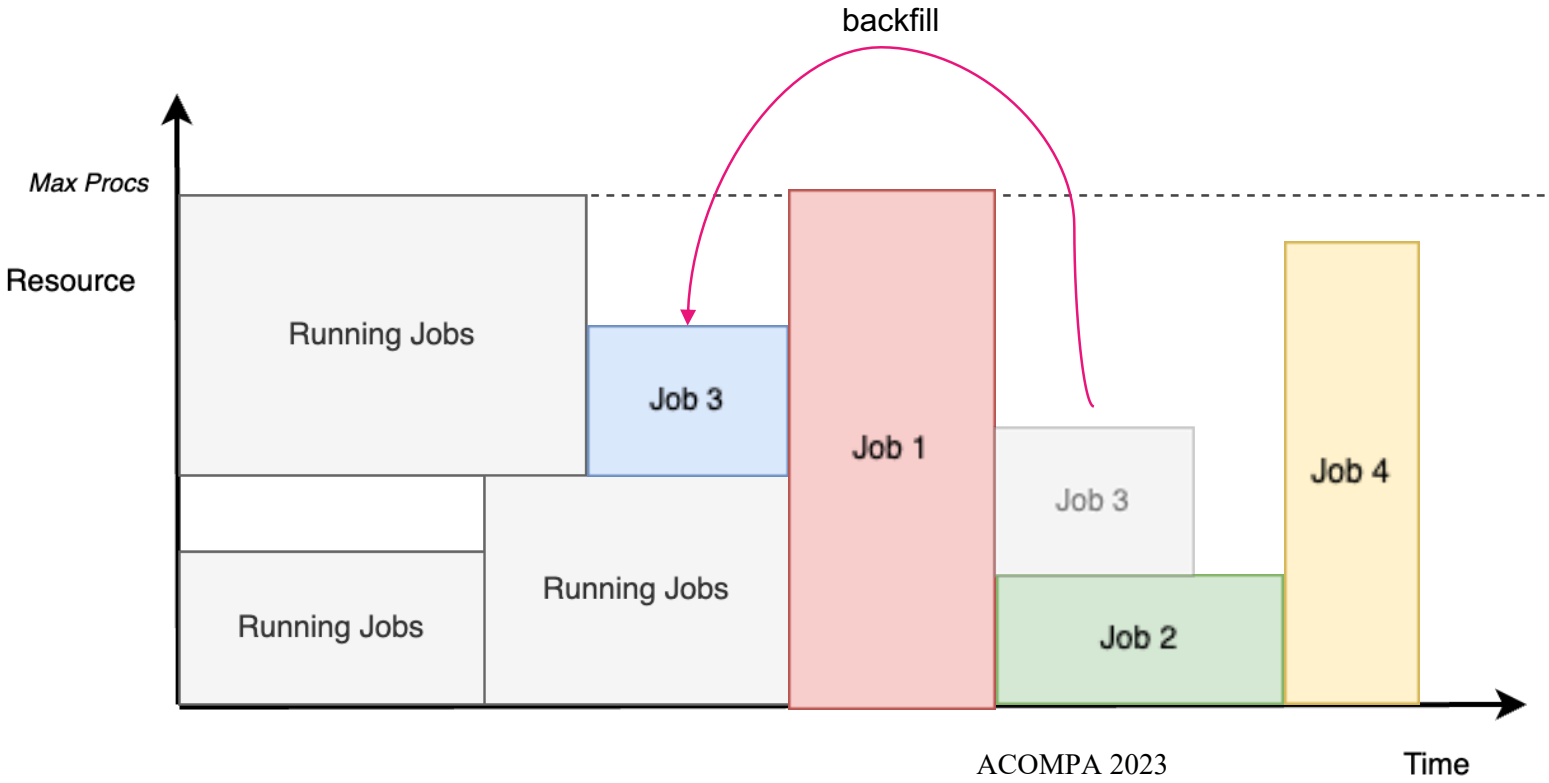
Scheduling on HPC Systems

Pure FCFS: Too many resource "holes"



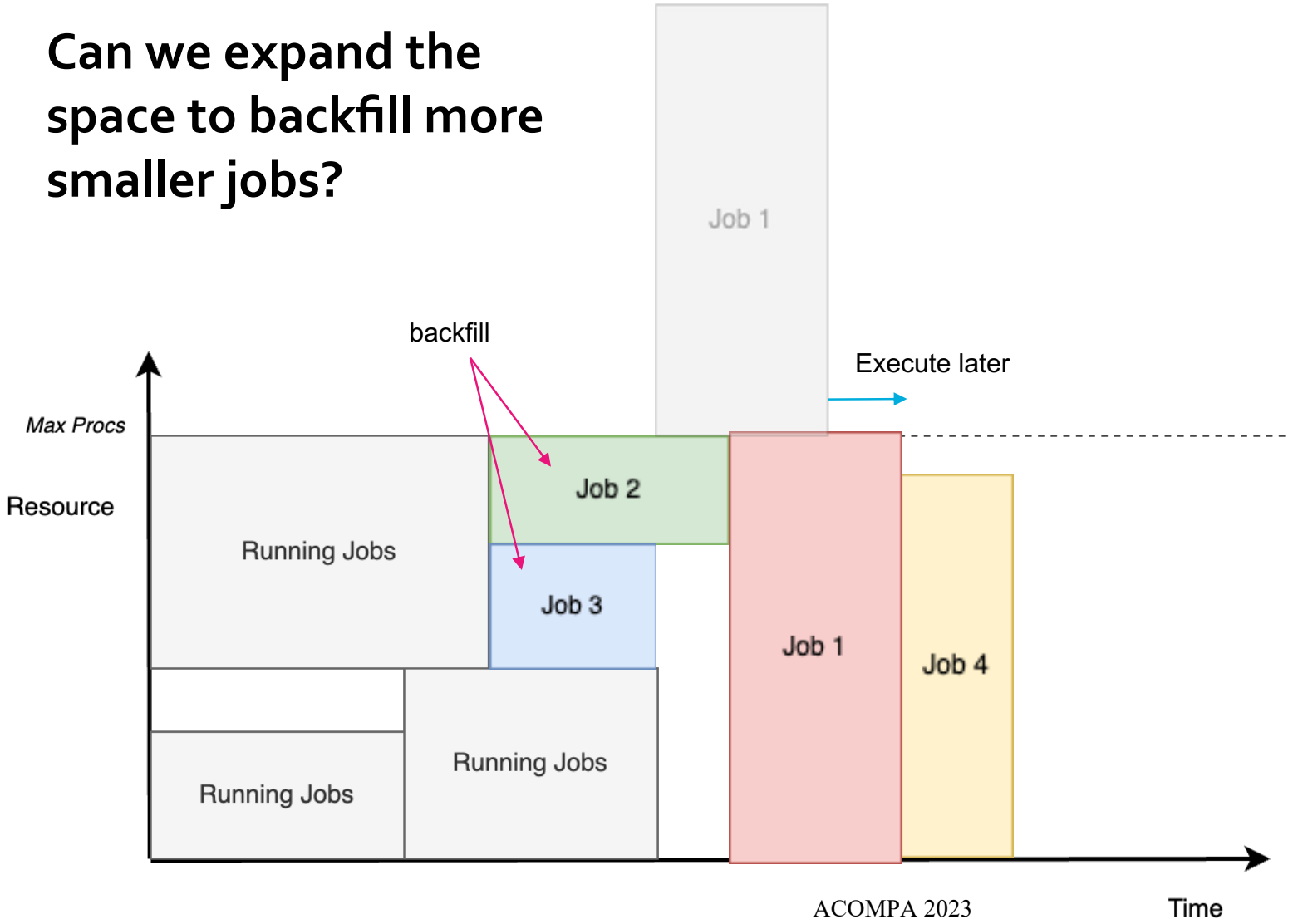
Scheduling on HPC Systems

FCFS with EASY Backfilling: allows smaller jobs to "backfill" the gaps left by the first job in the queue, as long as they don't delay the execution of the first job.



Scheduling on HPC Systems

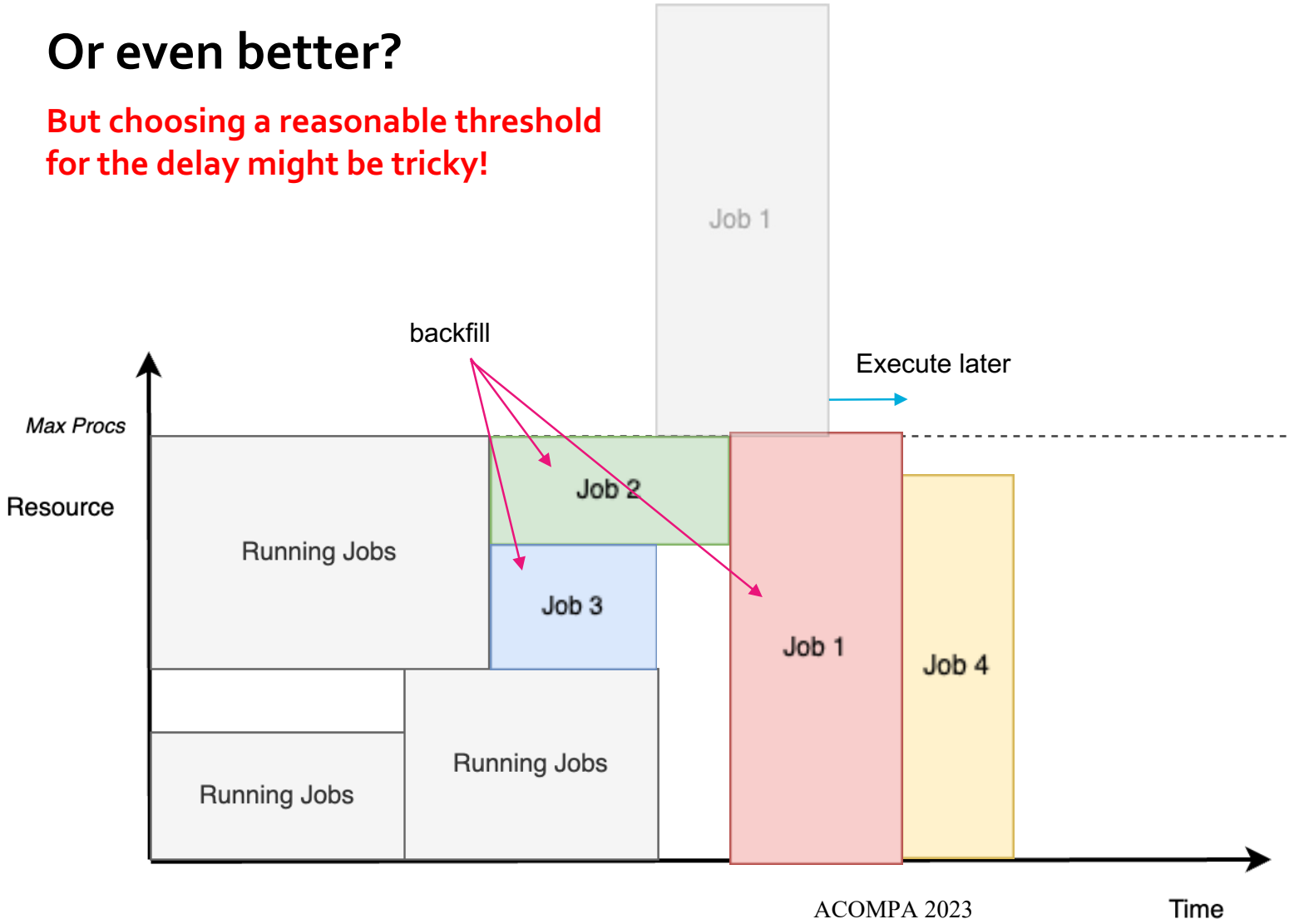
Can we expand the space to backfill more smaller jobs?



Scheduling on HPC Systems

Or even better?

But choosing a reasonable threshold for the delay might be tricky!

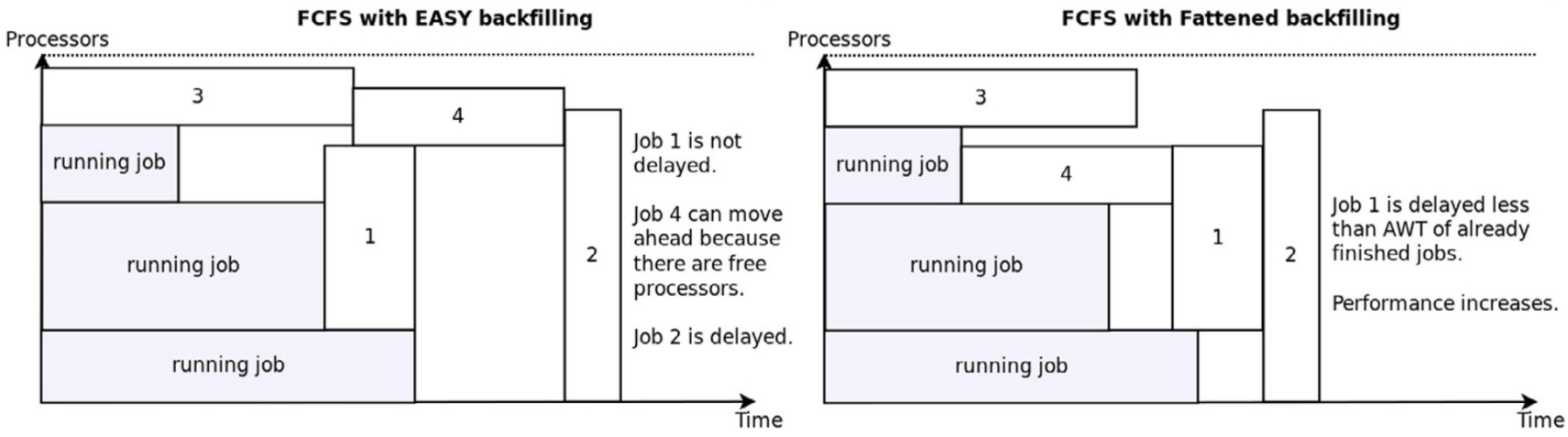


Research Goals

- Prove the potential of delaying queued jobs in improving the performance of real-world HPC systems.
- Propose a practical delay strategy that balances user fairness and scheduling performance.
- Develop a mechanism to encourage users to provide more accurate estimates

Related Work

Fattened backfilling is a recent job scheduling algorithm that provides more backfilling opportunities by allowing short jobs to move forward if they do not delay the first job of the queue more than the **average waiting time** (AWT) of the already finished jobs.



César Gómez-Martín, Miguel A. Vega-Rodríguez, José-Luis González-Sánchez, Fattened backfilling: An improved strategy for job scheduling in parallel systems, Journal of Parallel and Distributed Computing, Volume 97, 2016, Pages 69-77, ISSN 0743-7315,

Related Work

Algorithm 1 Pseudo-code of Fattened backfilling

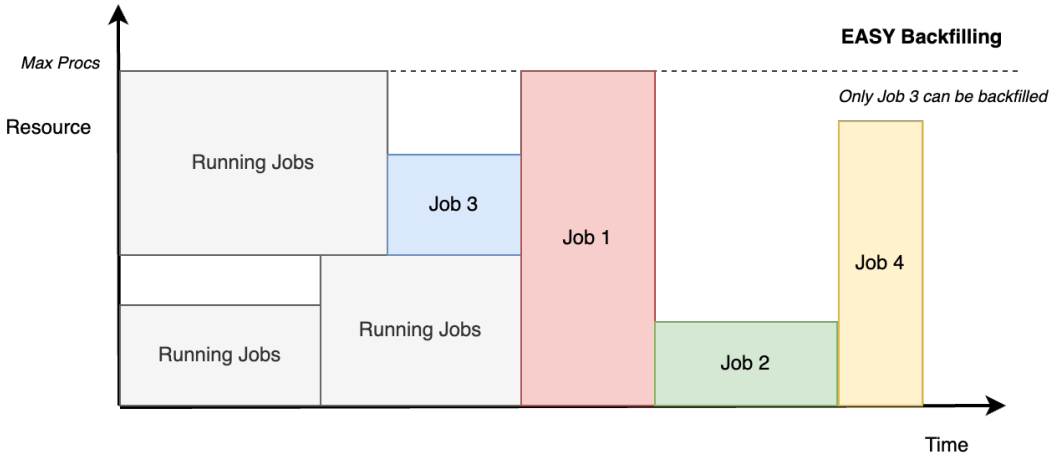
```
1:  function FATTENEDBACKFILLING
2:      totalWaitTime = 0;
3:      for all f in FinishedJobs F do
4:          totalWaitTime = totalWaitTime + (f.startTime − f.submitTime);
5:      AWT = totalWaitTime/F.size;
6:      for all r in RunningJobs R do
7:          Order r by expected termination time;
8:      for all j in OrderedRunningJobs J do
9:          Calculate the shadowTime in which the numberFreeCores is sufficient for the first queued job;
10:     for all q in OrderedQueuedJobs Q (already in order of arrival) do
11:         s = Q[0]; // First job of Q.
12:         if (q.requestedCores <= numberFreeCores)  &  (q.expectedFinishTime < (s.shadowTime + AWT)) then
13:             q is backfilled;
14:             break;
```

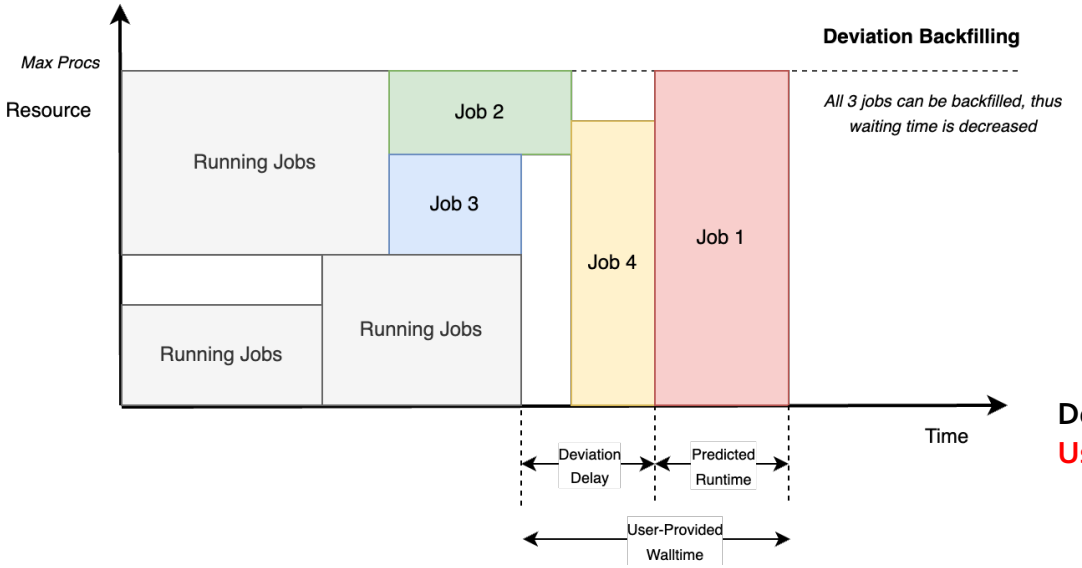
The first queued job is doubled their waiting time

(shadow time = current time + estimated wait time to have enough resources)

César Gómez-Martín, Miguel A. Vega-Rodríguez, José-Luis González-Sánchez, Fattened backfilling: An improved strategy for job scheduling in parallel systems, Journal of Parallel and Distributed Computing, Volume 97, 2016, Pages 69-77, ISSN 0743-7315,

Deviation Delay





Deviation Backfilling: The deviation between user estimates and system predictions is used as the delay threshold for the first job in the backfilling queue.

Deviation Delay =
User-Provided Walltime – Predicted Runtime

Deviation Delay

Why use the deviation of user-provided wall time?

- User estimates are usually much longer than the real execution time -> *More space!*
- Walltime acts as the "deadline" for each job. -> *Can delay as long as we do not cross this limit*

How to calculate the delay threshold?

Job Runtime Estimation

Use the ***k*NN** method!

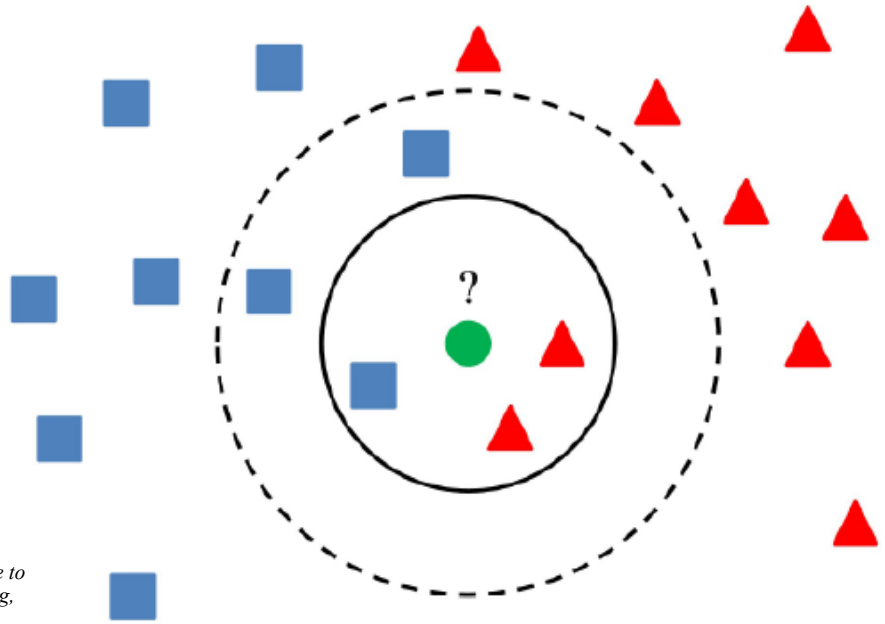
The inaccurate user estimate of a job is **refined** using the **historical data** about its most similar jobs

Predicted job runtime

User provided walltime

$$q^w = \frac{\widetilde{q^w}}{\Delta_q}$$

Predicted deviation ratio



T. H. Le Hai, L. L. Hoang, and N. Thoai, "Potential of applying kNN with soft walltime to improve scheduling performance," in 2021 The International Conference on Computing, Computational Modelling and Applications (ICCMA), 2021.

Evaluation & Discussion

- Experiments are conducted using BatSim
- Use 3 datasets (2 from Parallel Workload Archive and 1 from the SuperNode-XP):

HPC WORKLOADS STATISTIC

Workload Trace	From	Duration	#Jobs	#Nodes
SDSC-DS-2004	Mar 2004	13 months	96,069	171
ANL-Intrepid-2009	Sep 2009	8 months	68,936	640
SuperNode-XP-2017	May 2017	46 months	19,124	24

P.-F. Dutot, M. Mercier, M. Poquet, and O. Richard, "Batsim: a Realistic Language-Independent Resources and Jobs Management Systems Simulator," in 20th Workshop on Job Scheduling Strategies for Parallel Processing, Chicago, United States, May 2016.

D. G. Feitelson, D. Tsafrir, and D. Krakov, "Experience with using the parallel workloads archive," Journal of Parallel and Distributed Computing, vol. 74, no. 10, pp. 2967 – 2982, 2014.

Evaluation & Discussion

- Deviation Backfilling scheme outperforms other scheduling approaches

AVERAGE WAIT TIME RESULT

Scheduling Policy	SDSC-DS-2004	ANL-Intrepid-2009	SuperNode-XP-2017
EASY Backfilling	1098.2s	4982.7s	160158.1s
Fattened Backfilling	1095.9s	4253.7	156434.2s
Deviation Backfilling	**967.5s**	**4149.8s**	**140861.1s**
Perfect Estimate + EASY	1124.8s	4894.2s	121380.3s

Scheduling Policy	SDSC-DS-2004	ANL-Intrepid-2009	SuperNode-XP-2017
EASY Backfilling	15.58	16.16	1989.64
Fattened Backfilling	15.67	15.31	1917.88
Deviation Backfilling	**15.12**	**15.08**	**1546.62**
Perfect Estimate + EASY	15.12	14.35	857.07

Conclusion and future work

- Deviation Backfilling is a promising alternative to traditional scheduling algorithms, offering improved efficiency and user fairness.
- By using prediction deviation as the delay threshold for the first job in the backfilling queue, Deviation Backfilling can reduce the average waiting time for jobs and increase system throughput.
- Our work opens up several interesting directions for future research, including investigating the use of Deviation Backfilling in other types of HPC systems and exploring the use of other Machine Learning algorithms for job runtime prediction.

ACOMPA 2023

Thank you!

thanhhoang@hcmut.edu.vn

Thanh Hoang Le Hai, Khang Nguyen Duy,
Thin Nguyen Manh, Danh Mai Hoang and Nam Thoai

High Performance Computing Laboratory, Faculty of Computer Science & Engineering
Advanced Institute of Interdisciplinary Science and Technology
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City