Compositional Coordinated Resource Provisioning for Workflows with Stochastic Durations in Kubernetes

Tommaso Botarelli, Laura Carnevali, Leonardo Scommegna, Enrico Vicario

Dept. of Information Engineering, University of Florence

Software Technologies Lab - https://stlab.dinfo.unifi.it

QualITA'26, Florence, May 2026

This is about:

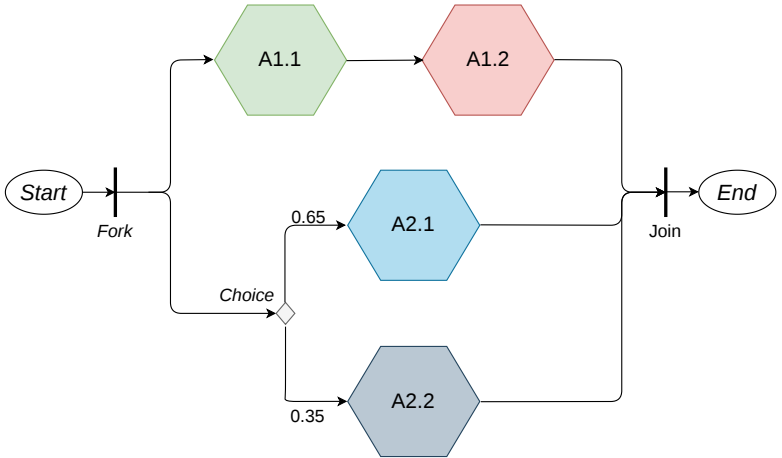
- Coordinated Resource Provisioning in Microservice Workflows
- End-to-End Response Time Optimization
- Service-specific and Topological information Combination
- Empirical In-Vitro Experiment on a Kubernetes Cluster

{ST LAB} Software Technologies Laboratory

UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

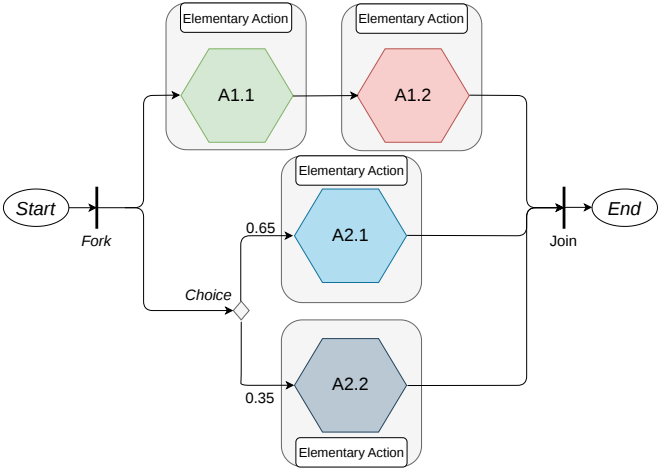
Workflow



- **Complex Functionalities:** multiple microservices interacting with each other

Software Technologies Laboratory

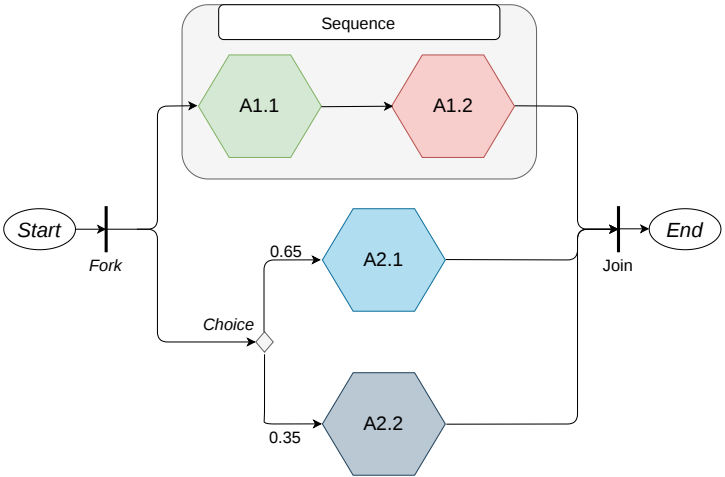
Simple Activities



- Microservices as atomic elements of **workflows**

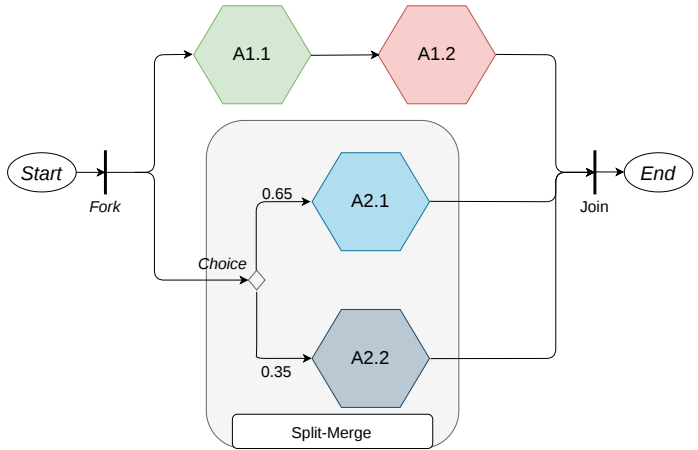
{ST LAB} Software Technologies Laboratory

Sequence



- Components executed **sequentially**

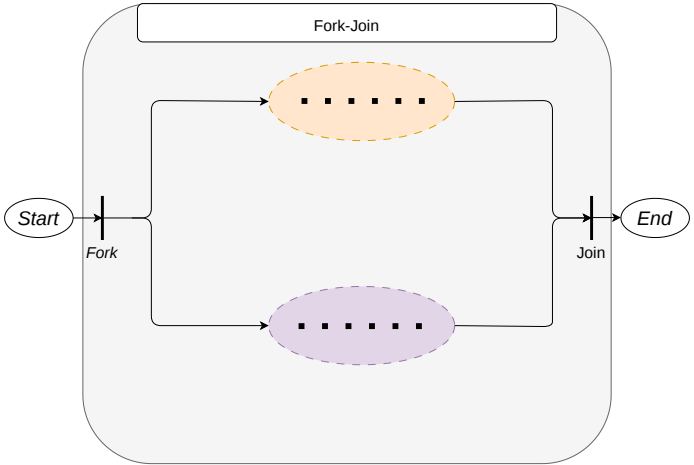
Split-Merge



- **Mutually exclusive** branch execution

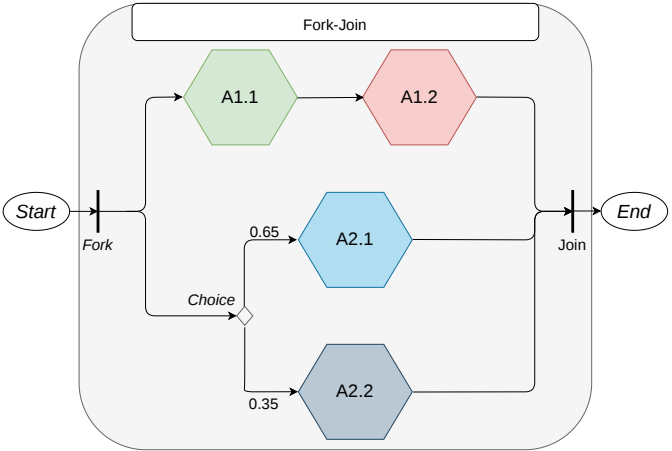
Software Technologies Laboratory

Fork-Join



- Branches execute **concurrently**

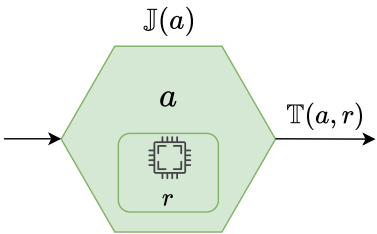
Fork-Join



- Branches execute **concurrently**

Completion Time of an Elementary Task

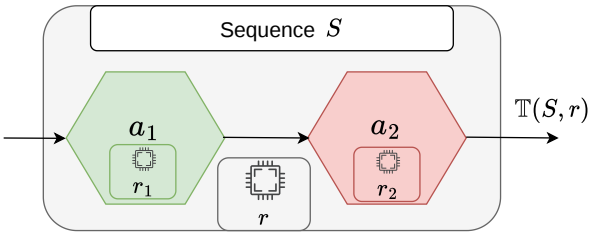
- r: Provisioned CPU Resources
- **Job Size**[1] $\mathbb{J}(a)$**:** Response Time with $r = 1$
- Response Time: $\mathbb{T}(a, r) = \frac{\mathbb{J}(a)}{r}$



[1] *Berg, Dorsman, Harchol-Balter "Towards Optimality in Parallel Scheduling" ACM Meas. Anal. Comput. Syst. 2017*

{ST LAB} Software Technologies Laboratory

Completion Time of a Sequence



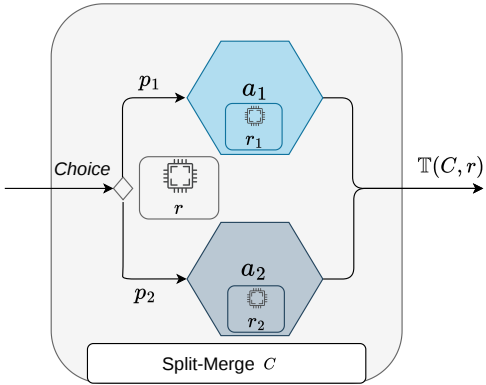
$$\mathbb{T}(S,r) = \mathbb{T}(a_1,r_1)+\mathbb{T}(a_2,r_2)$$

- S sequence of $a_1,\ldots,a_N$
- Response Time: $\mathbb{T}(a,r) = \sum_{n=1}^{N} \mathbb{T}(a_n,r_n)$
- $r = \sum_{n=1}^{N} r_n$
- a_i could be an elementary of **composite** task

{ST LAB} Software Technologies Laboratory

Completion Time of a Split-Merge

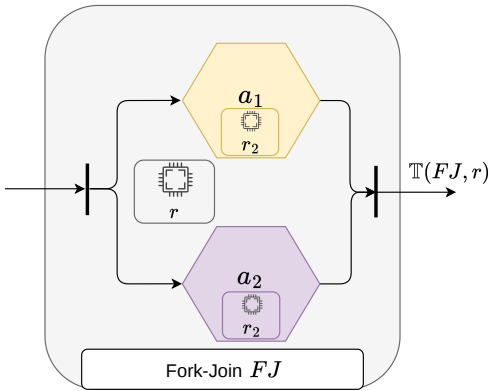
- C Choice of $a_1, \ldots, a_N$ with probabilities $p_1, \ldots, p_N$
- $\sum_{n=1}^{N} p_n = 1$
- $r = \sum_{n=1}^{N} r_n$
- $\mathbb{T}(a, r) = \sum_{n=1}^{N} p_n \cdot \mathbb{T}(a_n, r_n)$
- a_i could be an elementary of **composite** task



$$\mathbb{T}(C, r) = p_1 \cdot \mathbb{T}(a_1, r_1) + p2 \cdot \mathbb{T}(a_2, r_2)$$

Completion Time of a Fork-Join

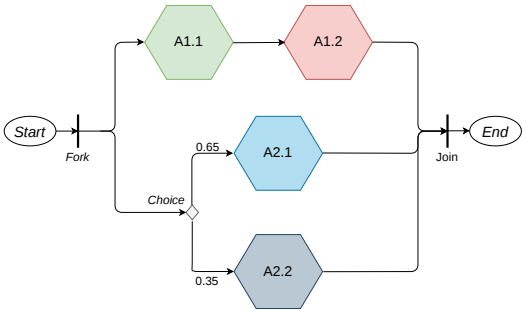
- FJ Fork-Join of $a_1, \ldots, a_N$
- $r = \sum_{n=1}^{N} r_n$
- $\mathbb{T}(a, r) = \max_{n=1}^{N} \mathbb{T}(a_n, r_n)$
- a_i could be an elementary of **composite** task



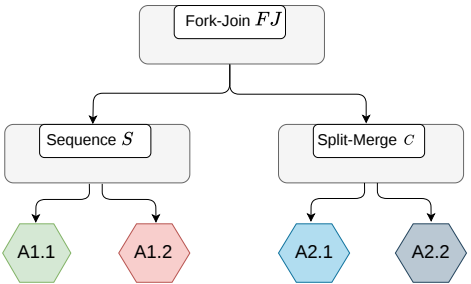
$\mathbb{T}(FJ, r) = max(\mathbb{T}(a_1, r_1), \mathbb{T}(a_2, r_2))$

{ST LAB} Software Technologies Laboratory

Structure Tree



Flat Representation

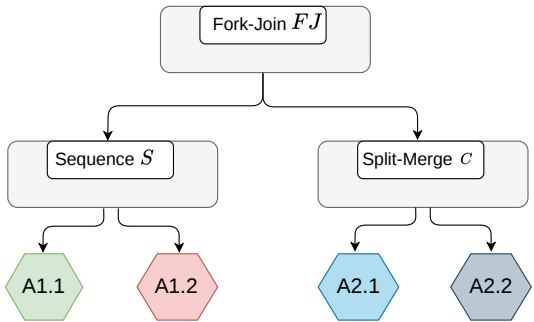


Structure Tree

- **E2E response Time**: $\mathbb{T}(FJ, r) = \max(\mathbb{T}(S, r_s), \mathbb{T}(C, r_s))$

{ST} LAB Software Technologies Laboratory

Problem Statement

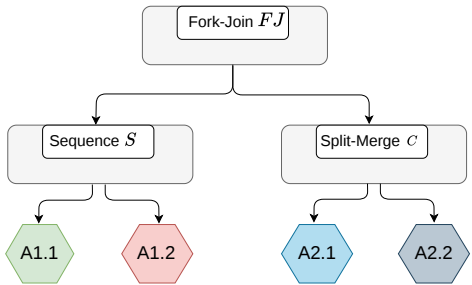


- End to End Reponse Time **depends** on:
  - Workflow Topology
  - Specific Elementary Activities (Job Sizes)
  - Resource Provisioning
- **Challenge**: provisioning of the elementary activities to optimize the E2E time

{ST LAB} Software Technologies Laboratory

Aim of the Work

- Evaluation of Compositional Coordinated Provisioning Strategy[2]
- **In-Vitro Experiment**:
  - Realistic workflows of microservices
  - Kubernetes Deployment
  - Evaluated against multiple abaltional strategies

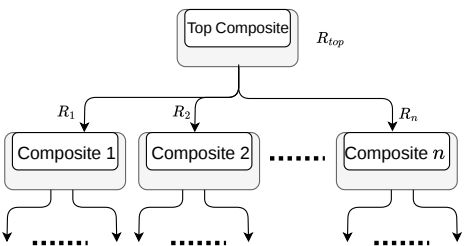


[2] *Carnevali, Paolieri, Reali, Scommegna, and Vicario "Compositional Coordinated Resource Provisioning in Workflows With Stochastic Durations", IEEE Trans. on Par. and Distrib. Sys., 2025*

{ST LAB} Software Technologies Laboratory

Compositional Coordinated Provisioning

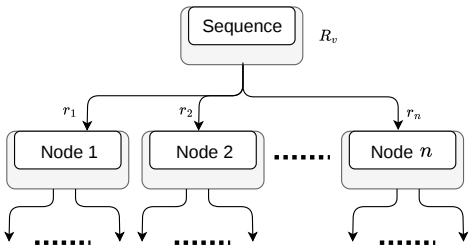
- **Top-Down** traversal of the Tree
- Traversal starts from the root node
- Root: budget equal to the total amount R
- Redistributing the assigned budget to child nodes
- Strategy adapted to the composition pattern (Sequence, Split, Fork)



{ST LAB} Software Technologies Laboratory

Sequence Provisioning Strategy

- Response-time **Proxy metric** $\gamma(\mathbb{T}(a, r))$
- R_v partitioned to **minimize** $\sum_{n=1}^{N} \gamma(\mathbb{T}(a_n, r_n))$
- **Constraint** $\sum_{n=1}^{N} r_n = R_v$
- Lagrangian formulation:

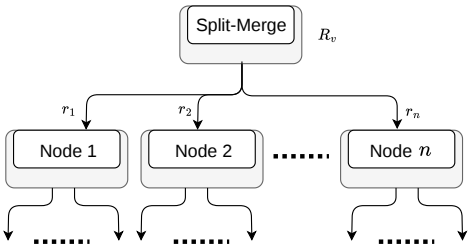


$$r'_n = \frac{\sqrt{\gamma(\mathbb{T}(a_n, r_n))}}{\sum_{i=1}^{N} \sqrt{\gamma(\mathbb{T}(a_n, r_i))}} \cdot R_v \qquad \forall\, n \in \{1, \ldots, N\}$$

{ST LAB} Software Technologies Laboratory

Split-Merge Provisioning Strategy

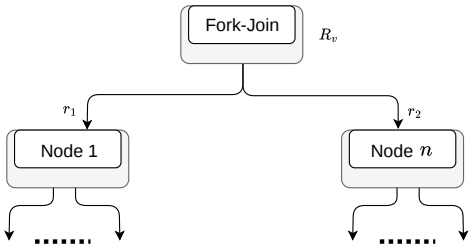
- Response-time **Proxy metric** $\gamma(\mathbb{T}(a, r))$
- R_v partitioned to **minimize** $\sum_{n=1}^{N} p_n \cdot \gamma(\mathbb{T}(a_n, r_n))$
- **Constraint** $\sum_{n=1}^{N} r_n = R_v$
- Lagrangian formulation :



$$r'_n = \frac{\sqrt{p_n \gamma(\mathbb{T}(a_n, r_n))}}{\sum_{i=1}^{N} \sqrt{p_i \gamma(\mathbb{T}(a_n, r_i))}} \cdot R_v \qquad \forall n \in \{1, \ldots, N\}$$

{ST LAB} Software Technologies Laboratory

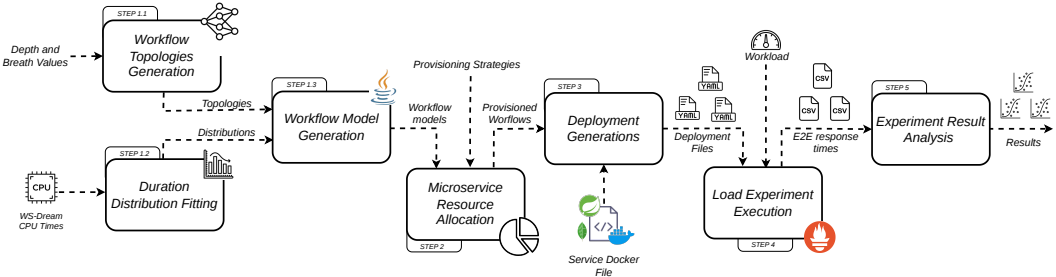
Fork-Join Provisioning Strategy



- Response-time **Proxy metric** $\gamma(\mathbb{T}(a, r))$
- R_v partitioned to **balance** E2E response time
- **Constraint** $\sum_{n=1}^{N} r_n = R_v$
- Heuristic for binary composition:

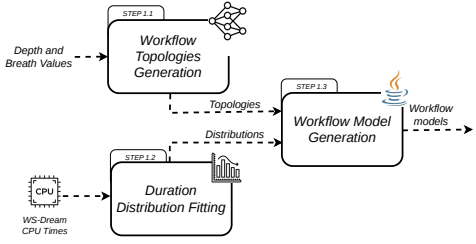
$$r'_n = \frac{\gamma(\mathbb{T}(a_n, r_n))}{\gamma(\mathbb{T}(a_n, r_n)) + \gamma(\mathbb{T}(a_{\bar{n}}, r_{\bar{n}}))} \cdot R_v \qquad n \in \{1, 2\}$$

Experimental Process



Experimental Process - Models Generation

- **Step 1.1 - Topologies Generation**:
  - Generated as tree structures using a top-down approach
  - Controlled Statistics: target tree height D and maximum branching factor B
  - 8 workflows generated using $D \in \{2, 3, 4\}$ and $B \in \{3, 4\}$
- **Step 1.2 - Durations Sampling**:
  - Elementary actions times sampled from WS-Dream dataset[3]
- **Step 1.3 Model Generation:**
  - Topologies and distributions used to create the structure tree

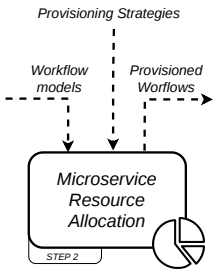


[3] *Zheng, Lyu "Ws-dream: A distributed reliability assessment mechanism for web services" 2008, DSN*

{ST LAB} Software Technologies Laboratory

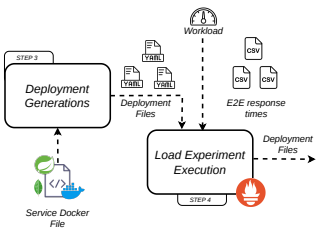
Experimental Process - Provisioning Calculation

- Comparison with three **alternative ablation strategies**
- **Same amount of resources**, different strategies
- **Completely Agnostic Provisioning (CAP)**:
  - Allocates R equally among all elementary tasks: $\forall n \in \{1, \ldots, N\}\ r_n = R/N$
- **Topology Driven Provisioning (TDP)**
  - Allocates R according to the workflow topology
  - Ignores information about the activity job size
- **Balanced Duration Driven Provisioning (BDDP)**
  - Allocates R according to the activity job size
  - Ignores information about the topology



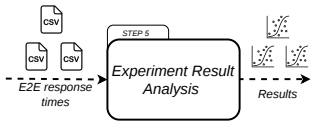
Experimental Process - Workflow Deployment

- **Deployment Generation:**
  - Automatically generated YAML files
  - Single services implemented with custom Docker Image
  - “Busy sleep” function for service operation
- **Load Experiment:**
  - **Low workload**: no concurrent requests
  - **High workload**: requests with exponential inter-arrival time with rate $\frac{4}{5}\mu$, with μ low workload avg response time
  - E2E response time collection



Experimental Process - Result Analysis

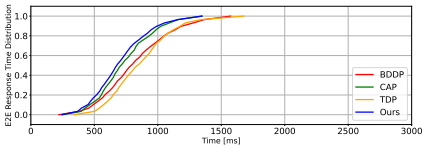
- Empirical PDF, $\widehat{f}(t)$, and CDF, $\widehat{F}(t)$, of E2E response time:
  - $\widehat{F}(t) = \frac{\#\text{samples} \leq t}{N}$ whit N total number of samples
  - $\widehat{f}(t)$ derived from ECDF
- **Pairwise Comparison Dominance**:
  - $\Delta(\tilde{\tau}, \tau) = D(\tilde{\tau}, \tau) - \frac{1}{2}$
  - $D(\tilde{\tau}, \tau) := \text{Prob}\{\tilde{\tau} \leq \tau\} = \int_0^\infty (1 - \widehat{F}_\tau(t)) \cdot \widehat{f}_{\tilde{\tau}}(t) dt$
  - If $\Delta(\tilde{\tau}, \tau) \in (0, \frac{1}{2}]$ then $\tilde{\tau}$ anticipates τ more than vice versa



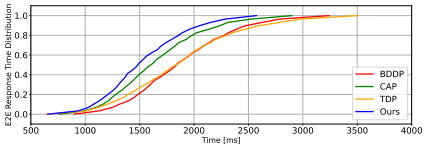
Results - Low Workload

workflow	structure	low-workload conditions $\Delta(\text{ours, CAP})$	$\Delta(\text{ours, TDP})$	$\Delta(\text{ours, BDDP})$
M1	D2-B3	0.0374	0.1796	0.1194
M2	D2-B3	0.0097	0.1569	0.2086
M3	D2-B4	0.0335	0.3073	0.1115
M4	D2-B4	0.0038	0.1439	0.1754
M5	D3-B3	0.0354	0.4136	0.2613
M6	D3-B3	0.0668	0.0241	0.2205
M7	D3-B4	0.0712	0.1809	0.2124
M8	D4-B3	0.0827	0.1637	0.1714
Δ Mean		0.0426	0.1963	0.1850

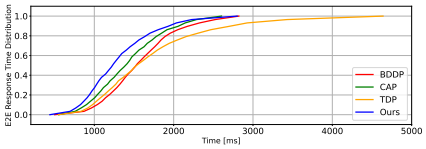
- Robust Consistent Superiority of our approach
- CAP performances could be related to:
  - How short jobs are scheduled in kubernetes
  - Behavior not guaranteed with greater skeew or durations



(a) Low workload - M1



(b) Low workload - M7

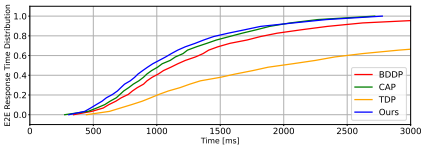


(c) Low workload - M8

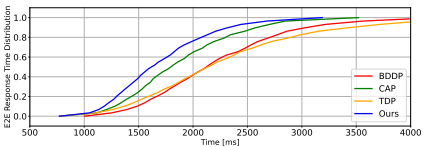
Results - High Workload

workflow	structure	low-workload conditions $\Delta(\text{ours, CAP})$	$\Delta(\text{ours, TDP})$	$\Delta(\text{ours, BDDP})$
M1	D2-B3	0.0363	0.2739	0.0923
M2	D2-B3	0.0414	0.2556	0.3740
M3	D2-B4	0.0210	0.4954	0.1416
M4	D2-B4	0.0632	0.4920	0.3533
M5	D3-B3	0.0164	0.4518	0.2791
M6	**D3-B3**	**0.0850**	**-0.0257**	**0.2074**
M7	D3-B4	0.0927	0.2183	0.2382
M8	D4-B3	0.0853	0.1809	0.2054
Δ Mean		0.0551	0.2928	0.2364

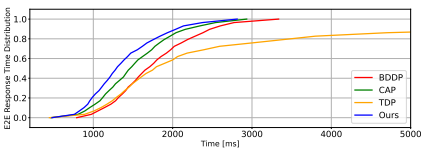
- Higher Dominance Deviation under high workload
- `M6` Corner case with high number of split-merges



(a) High workload - M1

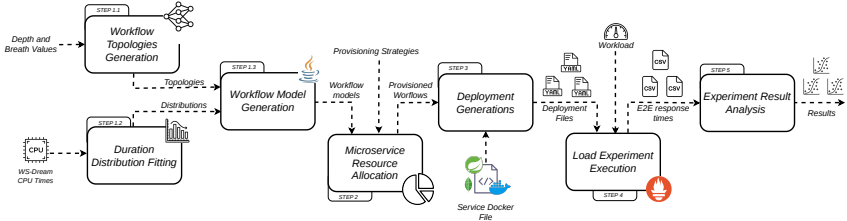


(b) High workload - M7



(c) High workload - M8

Conclusion



- Resource provisioning strategy combining workflow topology and processing time distributions
- Results show our approach consistently outperforms baselines, with its advantage amplifying under high-workload conditions
- Jointly analyzing topology and duration is essential to ensure stable and efficient resource provisioning
- **Future works**: Applying the coordinated methodology to variable workloads