Variation-Aware Resource Allocation Evaluation for Cloud Workflows using Statistical Model Checking

Saijie Huang, Mingsong Chen*, Xiao Liu,

Dehui Du and Xiaohong Chen

Shanghai Key Laboratory of Trustworthy Computing

East China Normal University

***Presenter**

華東師範大學
EAST CHINA NORMAL UNIVERSITY

Outline

- Introduction
- Preliminary Knowledge
  - Variation-aware NPTA
  - UPPAAL-SMC
- Our Quantitative TAS Evaluation Approach
  - Model Generation
  - Property Generation
  - SMC-Based Strategy Evaluation
- Experimental Results
- Conclusion

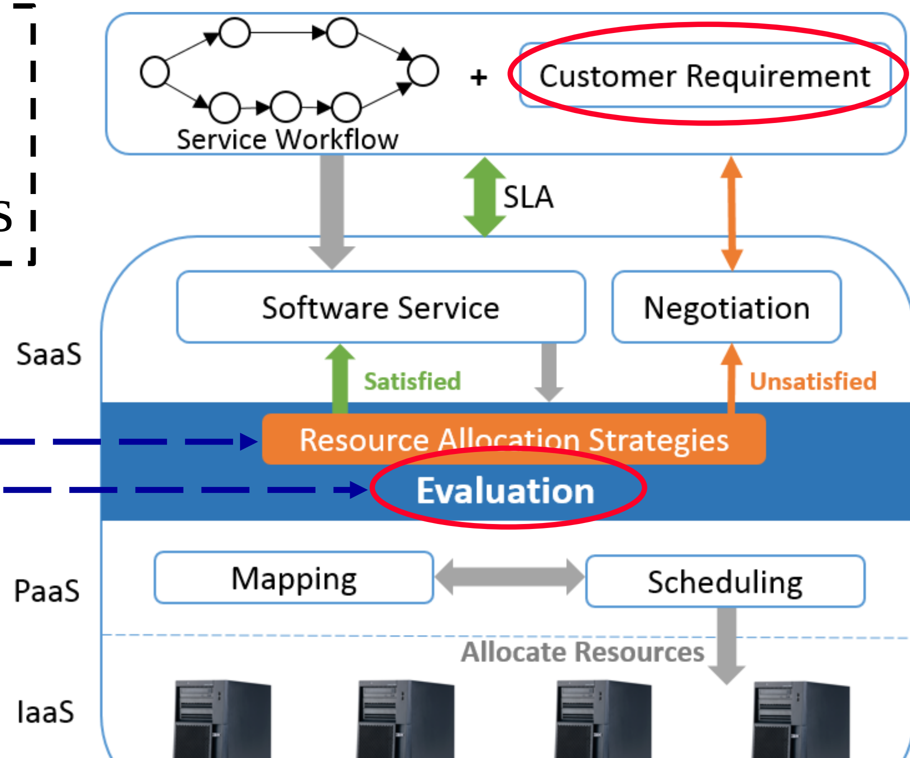
Outline

- Introduction
- Preliminary Knowledge
  - Variation-aware NPTA
  - UPPAAL-SMC
- Our Quantitative TAS Evaluation Approach
  - Model Generation
  - Property Generation
  - SMC-Based Strategy Evaluation
- Experimental Results
- Conclusion

Resource Allocation in Cloud Workflow

- Resource allocation is important for QoS of cloud workflow
  - ❑ Minimize service operating costs
  - ❑ Avoid Service Level Agreement (SLA) violations

Assign services to VMs considering various constraints



Requirements
- Cost
- Response time
- Reliability
- …

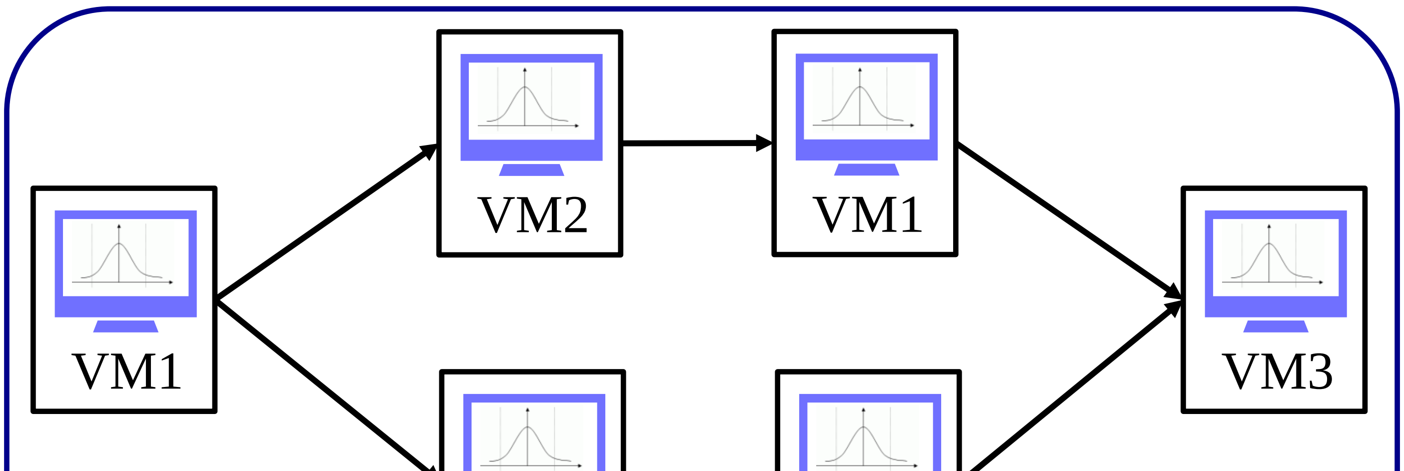
Is the reliability that the workflow can be completed in x hours with a cost of y dollars larger than z?

Evaluation tries to filter inferior

Resource allocation is an NP-complete problem!
Various approaches are proposed to find a optimal solution.

Challenges in Resource Allocation

- Due to execution variations, it's hard to determine which resource allocation strategy works best for a workflow coupled with QoS requirements.
  - E.g., time, cost and power consumption variation…



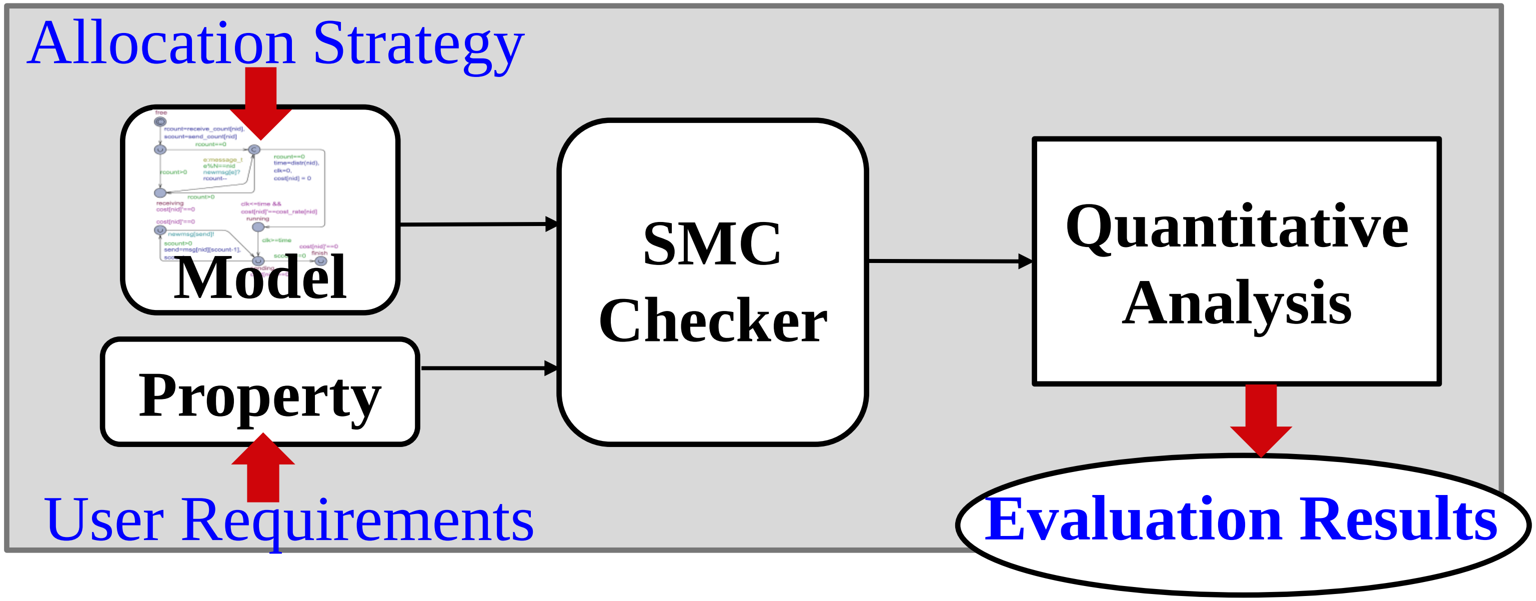
Challenges:

i) How to accurately model workflow-based services and customer requirements to enable the quantitative evaluation?

ii) How to model the time and cost variations caused by underlying infrastructures?

Statistical Model Checking (SMC)

- Our resource allocation is based on SMC, which is effective for checking large stochastic systems



- UPPAAL-SMC supported queries
  - Qualitative check: ***Pr [time <= bound] (<> expr) >= p***
  - Quantitative check: ***Pr [time <= bound] (<> expr)***
  - Probability comparison:

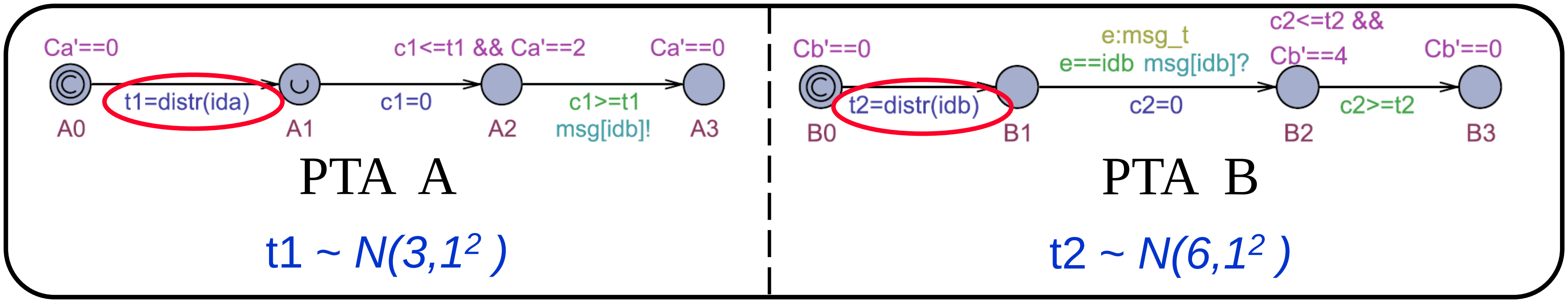
Pr [time1 <= bound1] (<> expr1) >= Pr [time2 <= bound2] (<> expr2)

Outline

- Introduction
- Preliminary Knowledge
  - Variation-aware NPTA
  - UPPAAL-SMC
- Our Quantitative TAS Evaluation Approach
  - Model Generation
  - Property Generation
  - SMC-Based Strategy Evaluation
- Experimental Results
- Conclusion

Variation-Aware NPTA

- NPTA - Network of Priced Timed Automata
- An NPTA instance, (A | B)



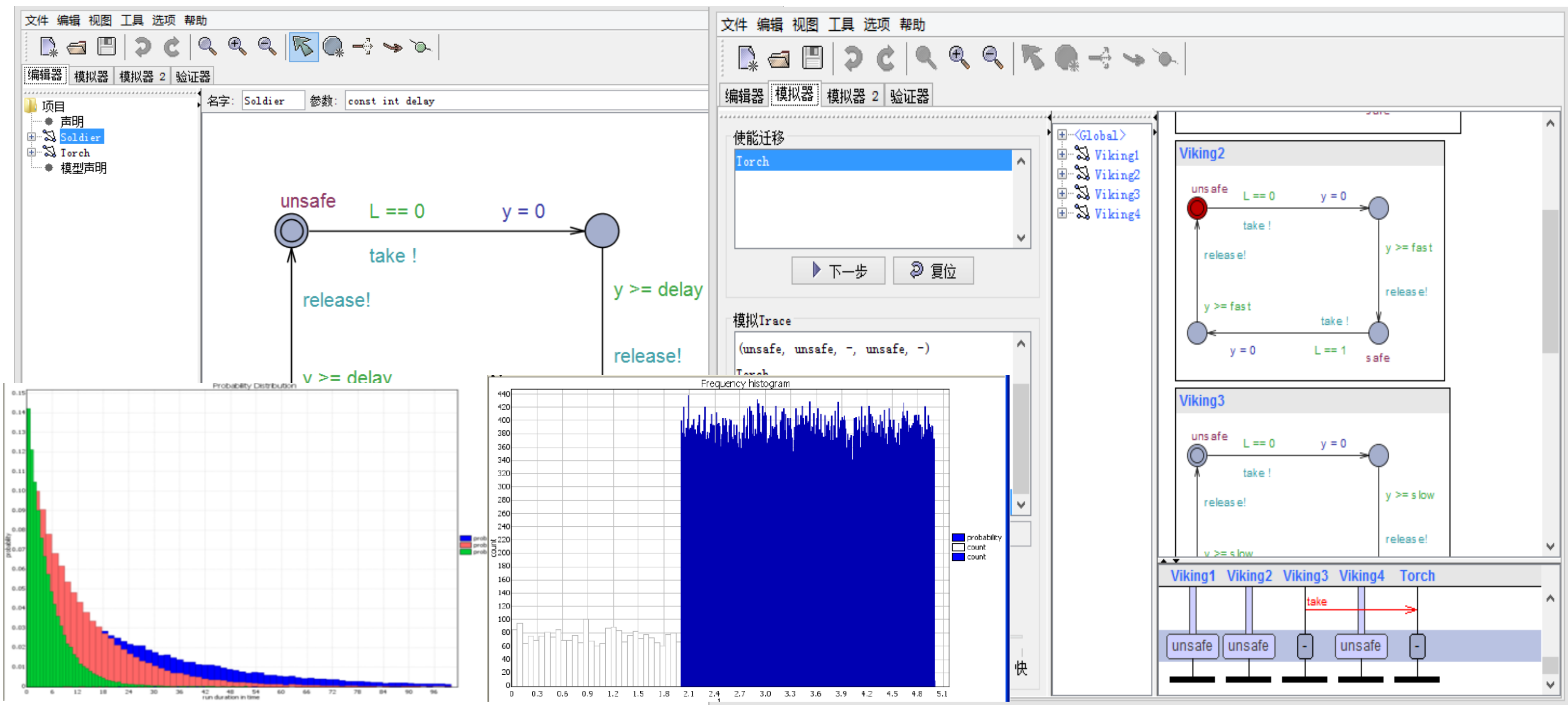
Time of reaching $(A3, B3) \sim N(9, 1^2+2^2)$.

- A possible behavior of the NPTA (A|B)

$$((A_0,B_0),[c_1=0,c_2=0,C_a=0,C_b=0]) \xrightarrow{0}$$
$$((A_1,B_1),[c_1=0,c_2=0,C_a=0,C_b=0]) \xrightarrow{0}$$
$$((A_2,B_1),[c_1=0,c_2=0,C_a=0,C_b=0]) \xrightarrow{2.5} \xrightarrow{msg[idb]!}$$
$$((A_3,B_2),[c_1=2.5,c_2=0,C_a=5,C_b=0]) \xrightarrow{5.1}$$
$$((A_3,B_3),[c_1=7.6,c_2=5.1,C_a=5,C_b=20.4]) \xrightarrow{\cdots} \ldots$$

UPPAAL-SMC

- SMC versus formal model checking
  - Based on simulation, thus requiring far less memory and time
  - Allow high scalable validation approximation
  - Support quantitative performance analysis
- Application Scenarios: Biology, energy-aware buildings…

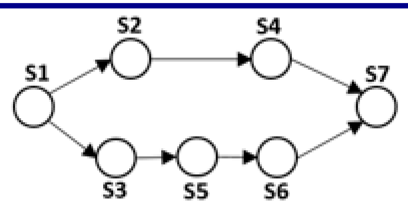


Outline

- Introduction
- Preliminary Knowledge
  - Variation-aware NPTA
  - UPPAAL-SMC
- Our Quantitative TAS Evaluation Approach
  - Model Generation
  - Property Generation
  - SMC-Based Strategy Evaluation
- Experimental Results
- Conclusion

Problem Definition

DAG *G = (V,E)*



Variation Information

Execution time variation
+ Cost variation

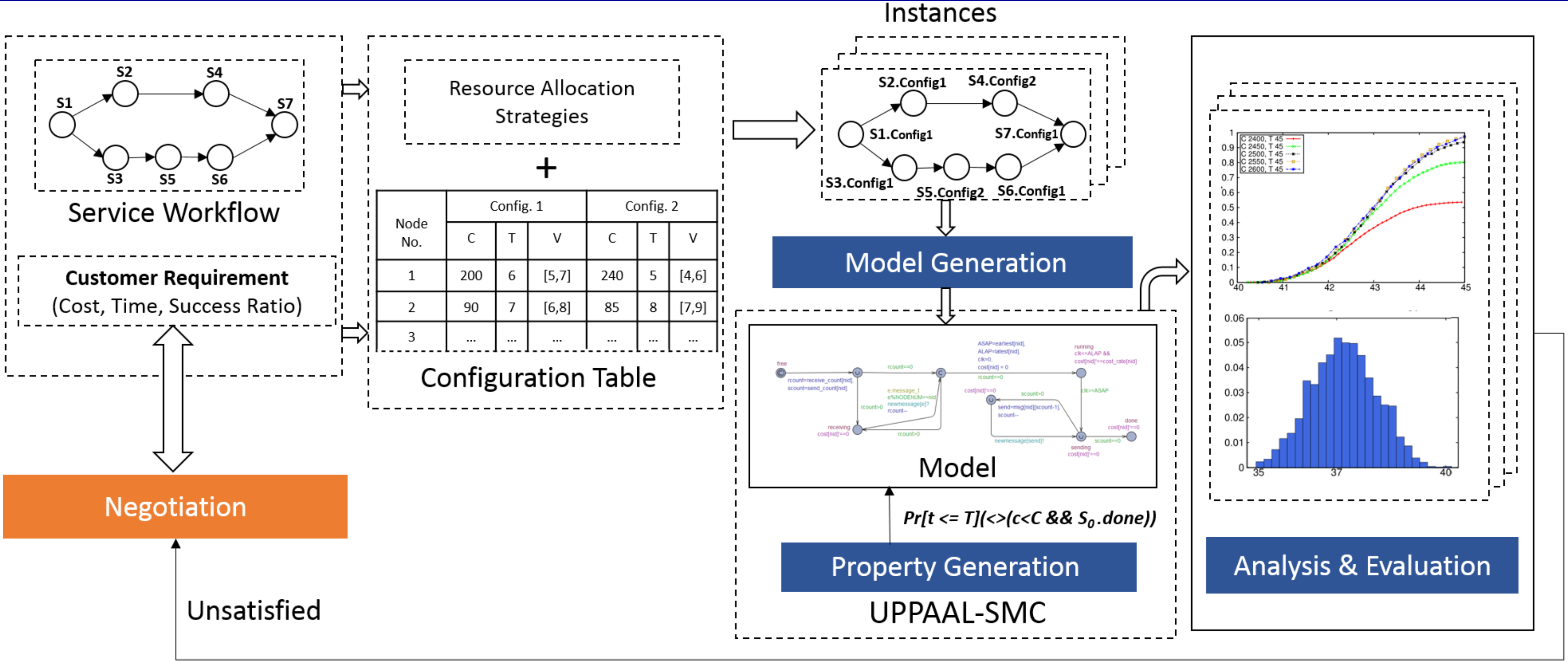
Customer Requirement *R*(*C*,*T*,*SR*)

- Response Time
- Cost
- Success ratio

Quantitative Evaluation Framework

Quantitative Analysis Results

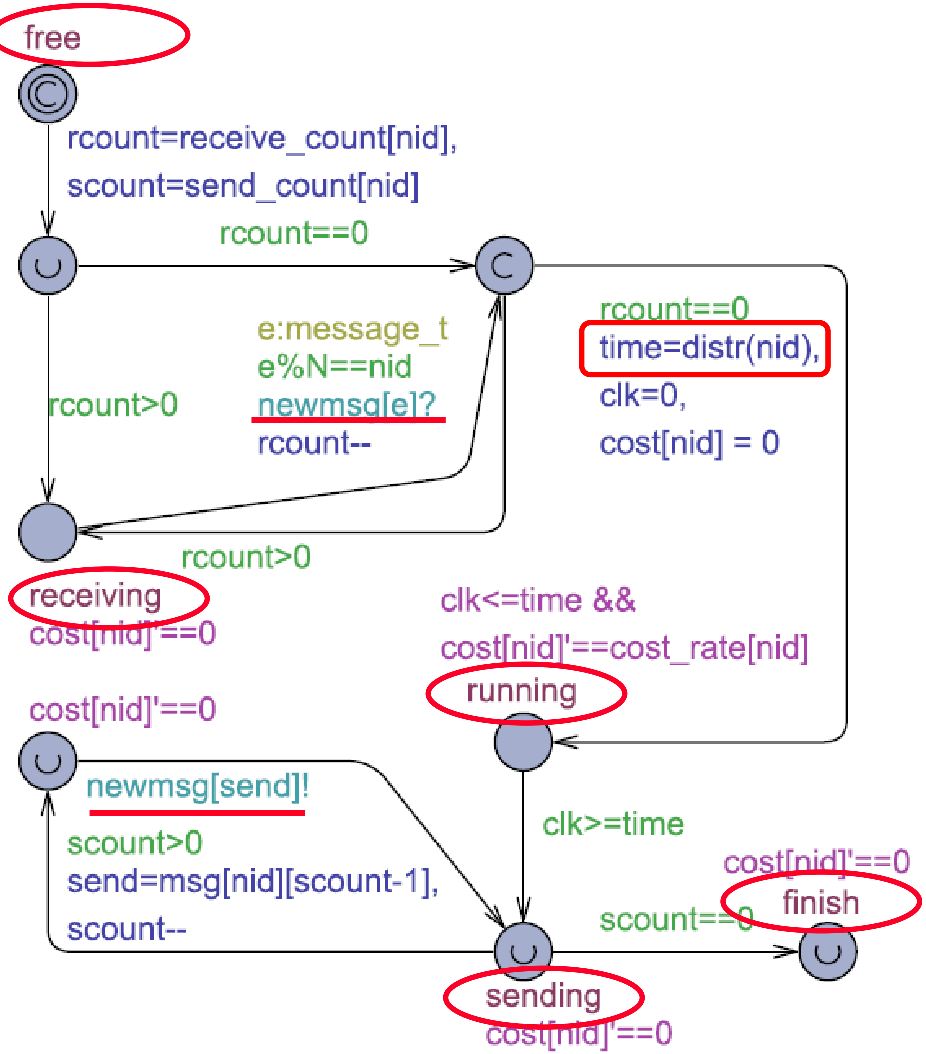
Our Framework



- **Model Generation:**
  - Resource allocation instances are translated into NPTA model
- **Property Generation**
  - Customer requirements are converted into propoerties to enable queires
- **Analysis & Evaluation**
  - Conduct the automated quantitative analysis using UPPAAL-SMCs

NPTA Model Generation

Front-end Model

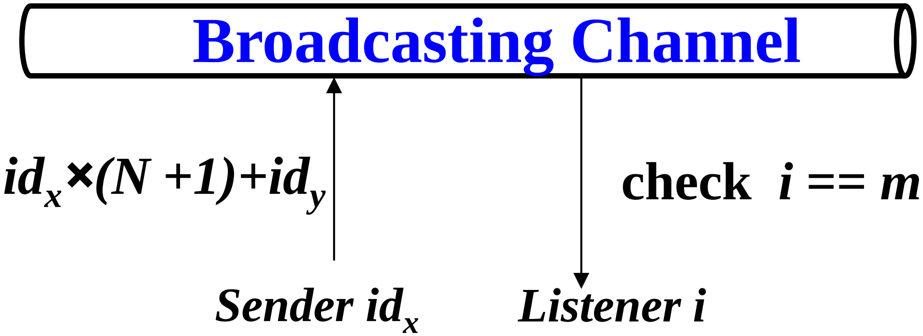


- Free state
  - The beginning of a service
- Receiving state
  - Tries to obtain notification messages from all the predecessors
- Running state
  - All predecessors finished
  - Current service is executing
- Sending state
  - Notify all successive services about its completion
- Finish state
  - Indicate the completion of a service

NPTA Model Generation

Back-end Configuration describes both concurrent semantics of workflows and execution variation information.

- **Workflow configuration**
  - To use workflow matrix *msg* to indicate workflow edges
  - *msg[i][j]=1*: message sent from i[th] service to j[th] service
- **Variation configuration**
  - Describe the time distributions of services
  - *distribution[N+1][2]:*store expected time and standard deviation
- **Communication between services**



$encode\_msg(id_x, id_y) = id_x \times (N+1) + id_y$

check $i == m\%(N + 1)$

Property Generation

"What is the probability that the workflow can be completed using a time of x with a cost of y?"

Pr[<= x](<> (cost[1]+. . .+cost[N]) <= y && System.done)

- ***[<= x]*** **indicates the time constraint of the cloud workflow**
- ***<>p checks*** **whether customer requirement p can be fulfilled eventually.**
- ***System.done*** **indicates the completion of the whole workflow**
- ***(cost[1]+. . .+cost[N]) <= y*** **indicates the overall cost of the workflow execution cannot be larger than** ***y***

Resource Allocation Instance Generation

Our framework has 3 built-in resource allocation heuristics

- **Cost-Constraint Time Minimization(CCTM)**
  - **Search a time optimal instance while the cost constraint is satisfied**
- **Time-Constraint Cost Minimization (TCCM)**
  - **Search a cost optimal instance while the time constraint is not violated**
- **xth-Round Feasible Instance (xRFI)**
  - **The xth feasible resource allocation instance encountered in the exhaustive enumeration**

Resource Allocation Strategy Evaluation

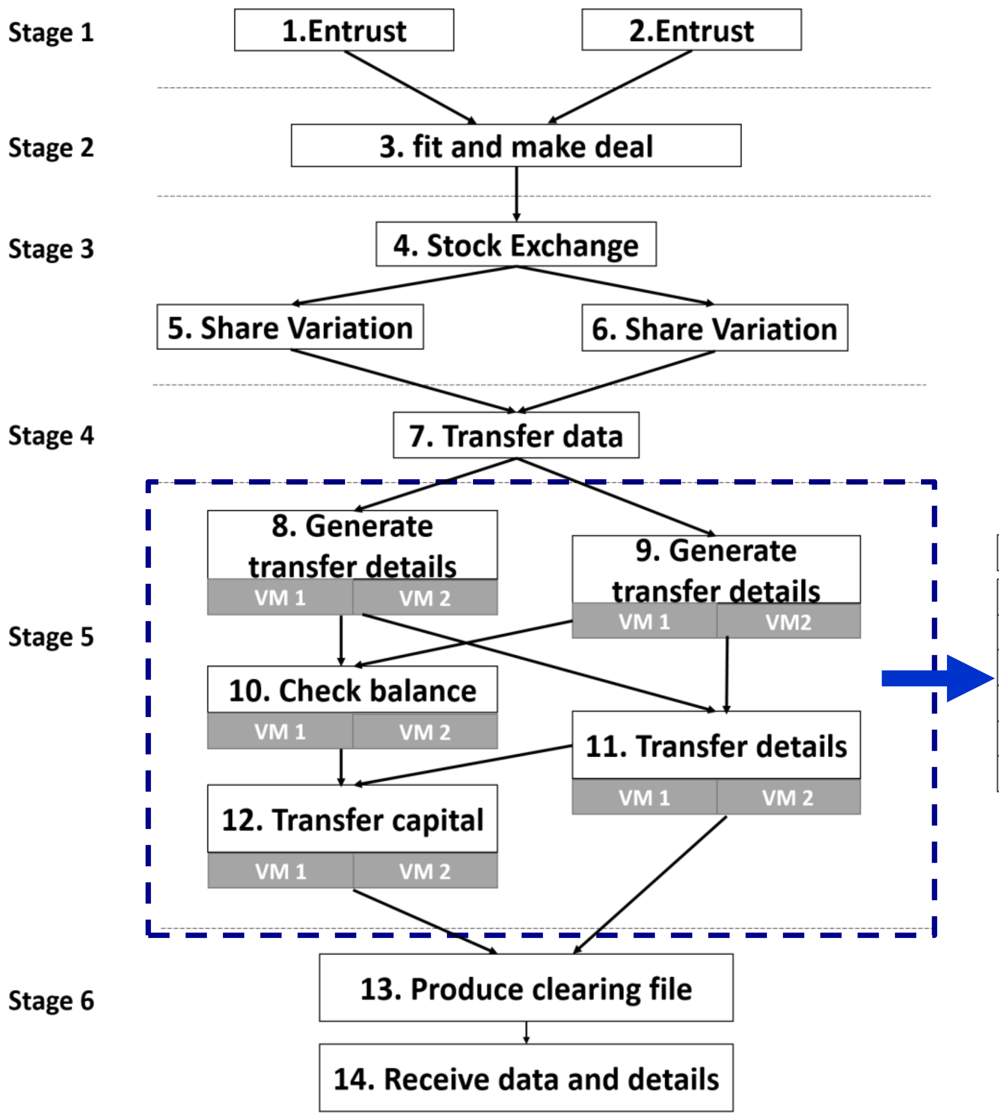
Our framework supports 3 evaluation strategies

- Single Requirement Multiple Strategies (SRMS)
  - **SRMS tries to compare multiple strategies and filter inferior ones**
- Multiple Requirements Single Strategy (MRSS)
  - **MRSS tries to tune the parameters of the strategy to achieve a better performance**
- Multiple Requirements Multiple Strategies (MRMS)
  - **MRMS supports both inferior solution filtering and parameter tuning**

Outline

- Introduction
- Preliminary Knowledge
  - Variation-aware NPTA
  - UPPAAL-SMC
- Our Quantitative TAS Evaluation Approach
  - Model Generation
  - Property Generation
  - SMC-Based Strategy Evaluation
- Experimental Results
- Conclusion

Case Study



A cloud workflow of
Shanghai A-Share Stock Market

For node 8 to node 12:

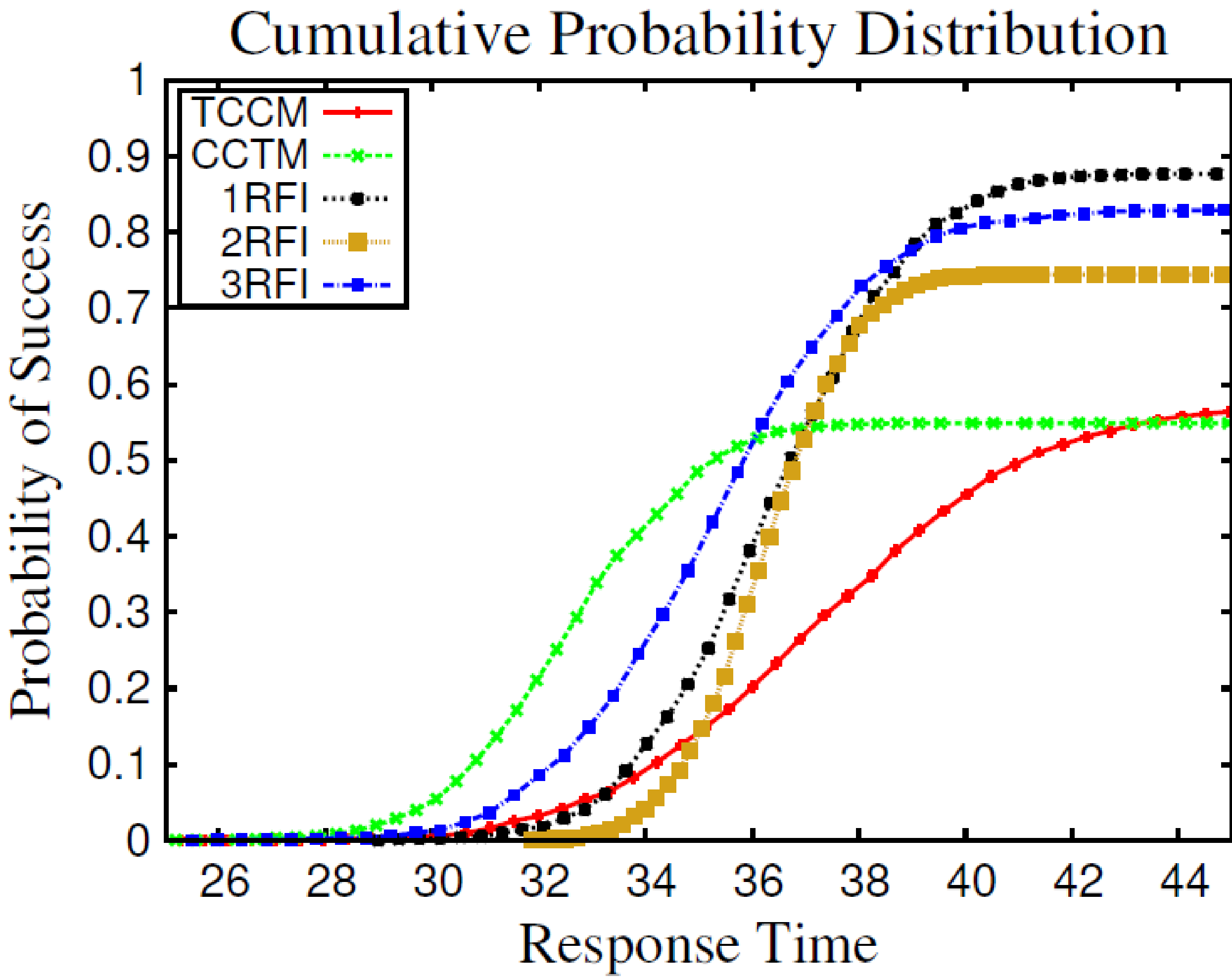
	Config. 1			Config. 2		
Node	Price	M.T.(μ)	S.D.(σ)	Price	M.T.(μ)	S.D.(σ)
8	70	5	0.4	50	6	0.7
9	45	10	0.8	60	8	0.5
10	40	10	0.5	30	12	0.8
11	100	7	0.6	80	8	0.9
12	60	12	0.4	40	15	0.7

* VMs with higher price is faster and more stable.

SRMS Approach

Customer Requirements: Completed within 45 time units and 2550 cost units, and the success ratio to be no lower than 80%.

Pr[<= 45](<> (cost[1]+. . .+cost[N]) <= 2550 && S0.done)



1RFI strategy selected!

Both results of TCCM and CCTM are less than 0.6 !

MRMS Approach

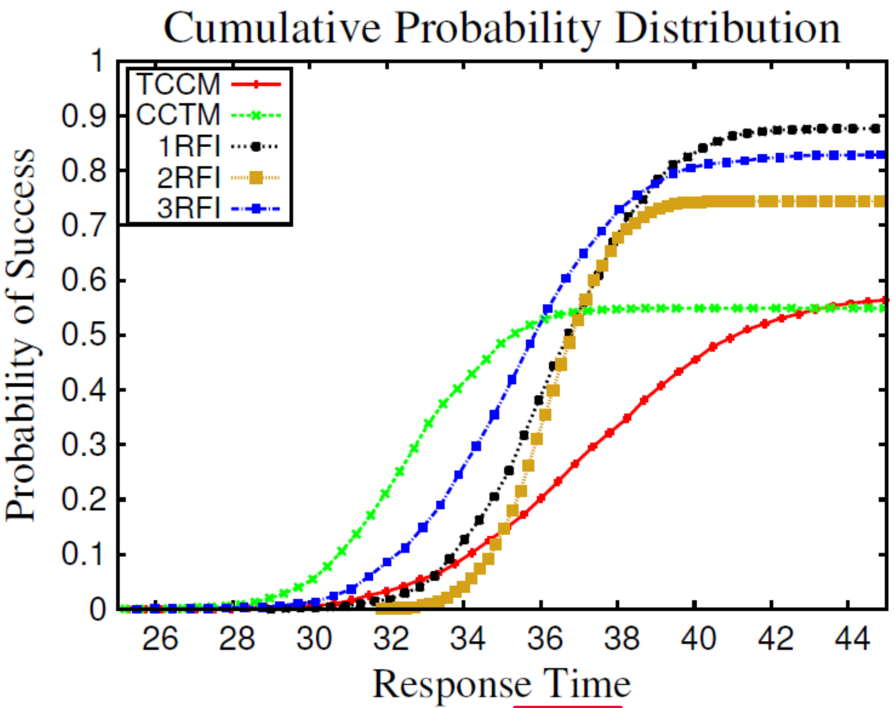


Fig. 1 CPD for R(2550, 45, 80%)

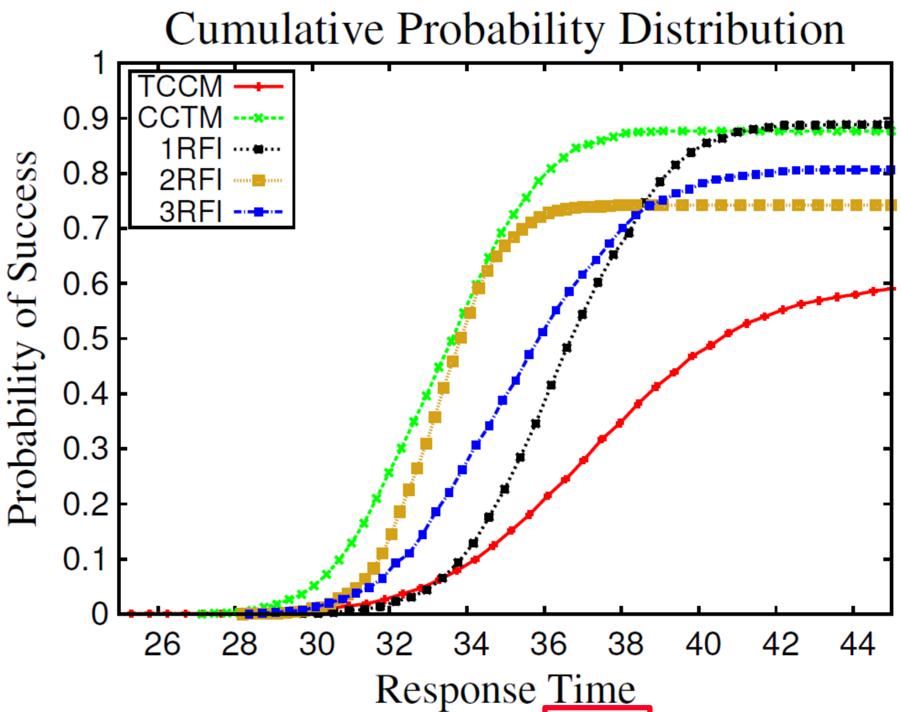


Fig. 2 CPD for R(2600, 45, 80%)

- Tune the cost constraint from 2550 to 2600
  - CCTM can achieve better success ratio as price increased, since workflow can get better VMs.
  - 2RFI's response time performance improved.

MRMS Approach

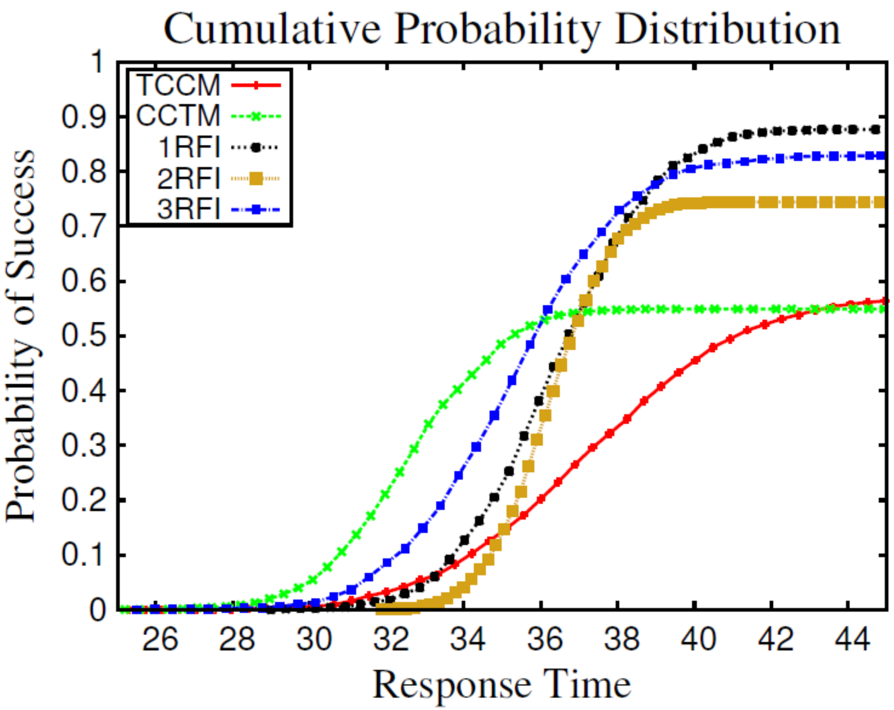


Fig. 1 CPD for R(2550, 45, 80%)

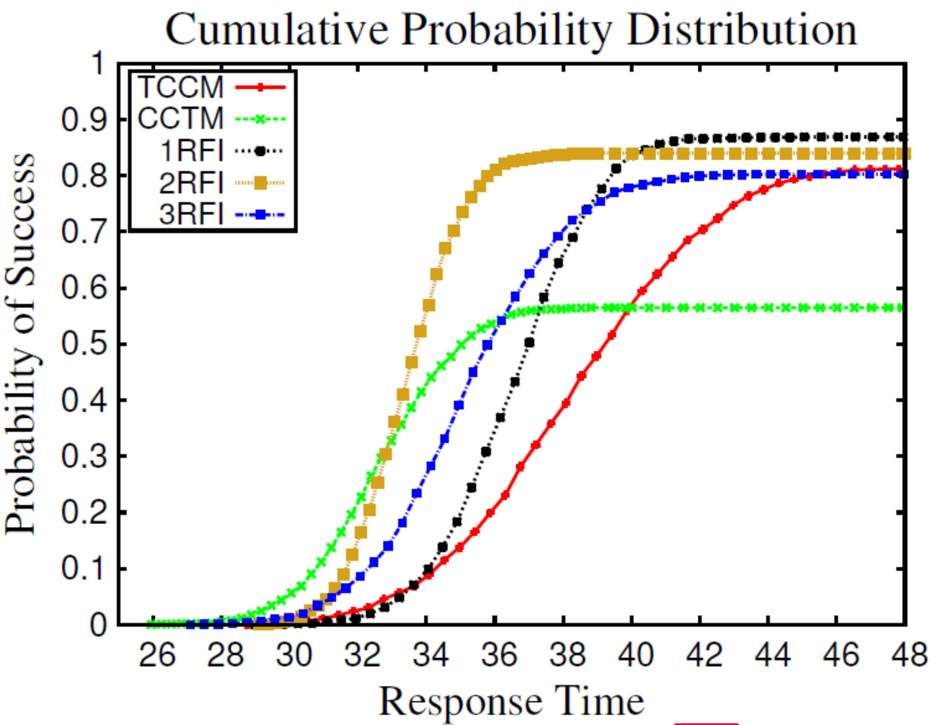
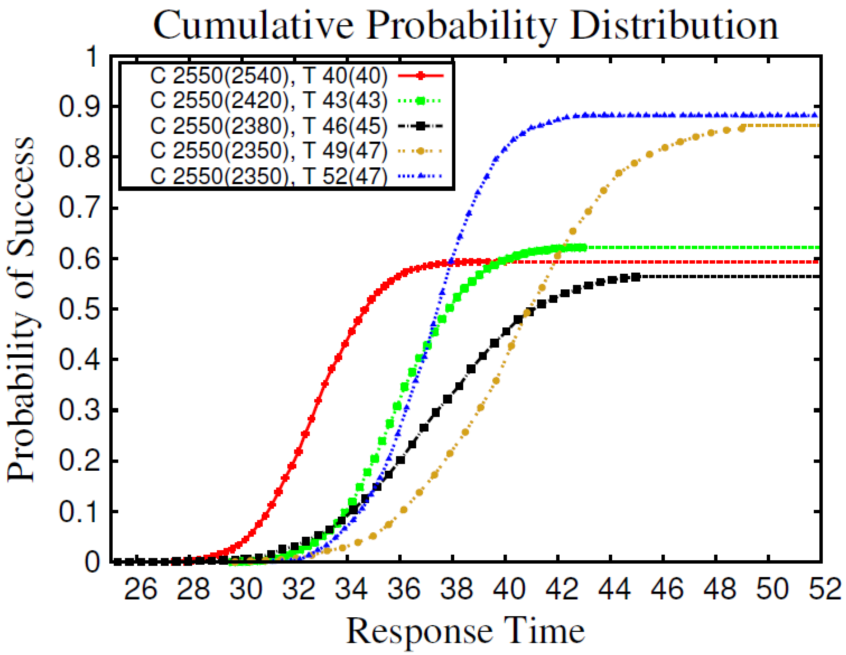


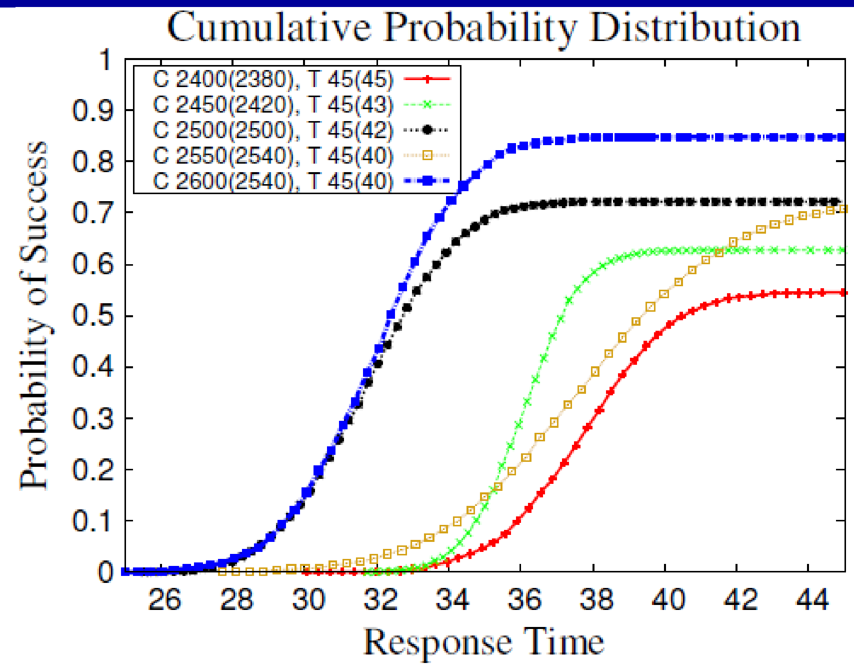
Fig. 3 CPD for R(2550, 48, 80%)

- Tune the time constraint from 45 to 48
  - Significant increase of success ratio for instances TCCM and 2RFI
  - In Fig.3, 2RFI instance has a better response time
  - 1RFI has the best probability of success in both cases

MRSS Approach



CPD for TCCM Strategy
Tune the time only!



CPD for CCTM Strategy
Tune the cost only!

Due to the relaxed time constraint, the success ratio is drastically improved.

Increase of the cost lead to the reduction of response time.

Conclusion

- Variation-aware resource allocation is important for the QoS of cloud workflow
  - Reduce overall operating costs & SLA violation
- Propose a an SMC-based evaluation framework
  - Support complex QoS queries to filter inferior resource allocation solutions
  - Enable tuning of QoS constraints
- Successfully apply our approach on an industry cloud workflow
  - Demonstrate the effectiveness of our framework

Thank you !