

# Efficient QoS-aware Service Composition with a Probabilistic Service Selection Policy

Adrian Klein

**The University of Tokyo**

1<sup>st</sup> year PhD student

[www.adrianobits.de](http://www.adrianobits.de)



# What **PROBLEM** we tackle

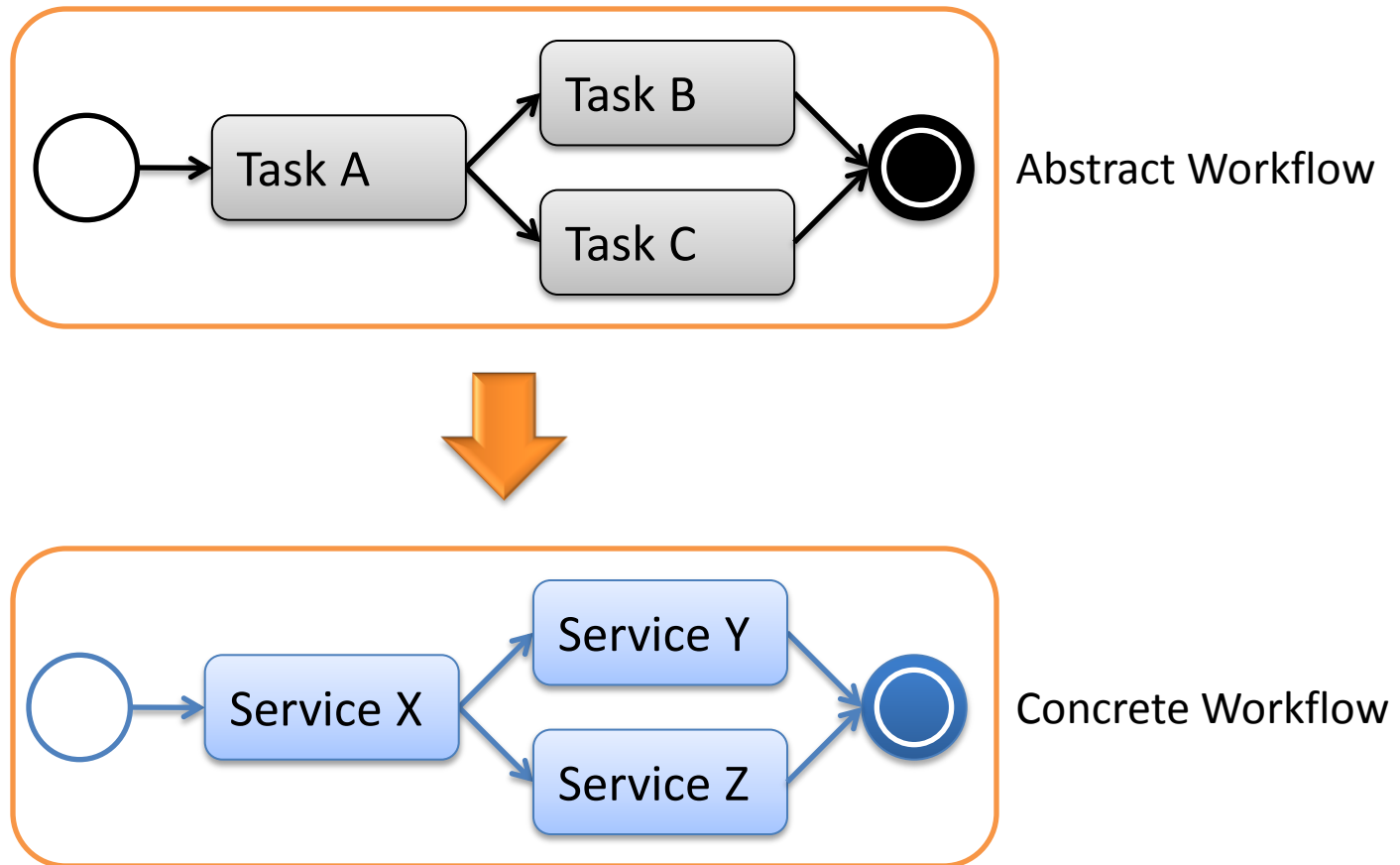


within the **Service Composition...**

# Service Composition:

Given an **abstract** workflow,  
select **concrete** services to execute it.

# Service Composition

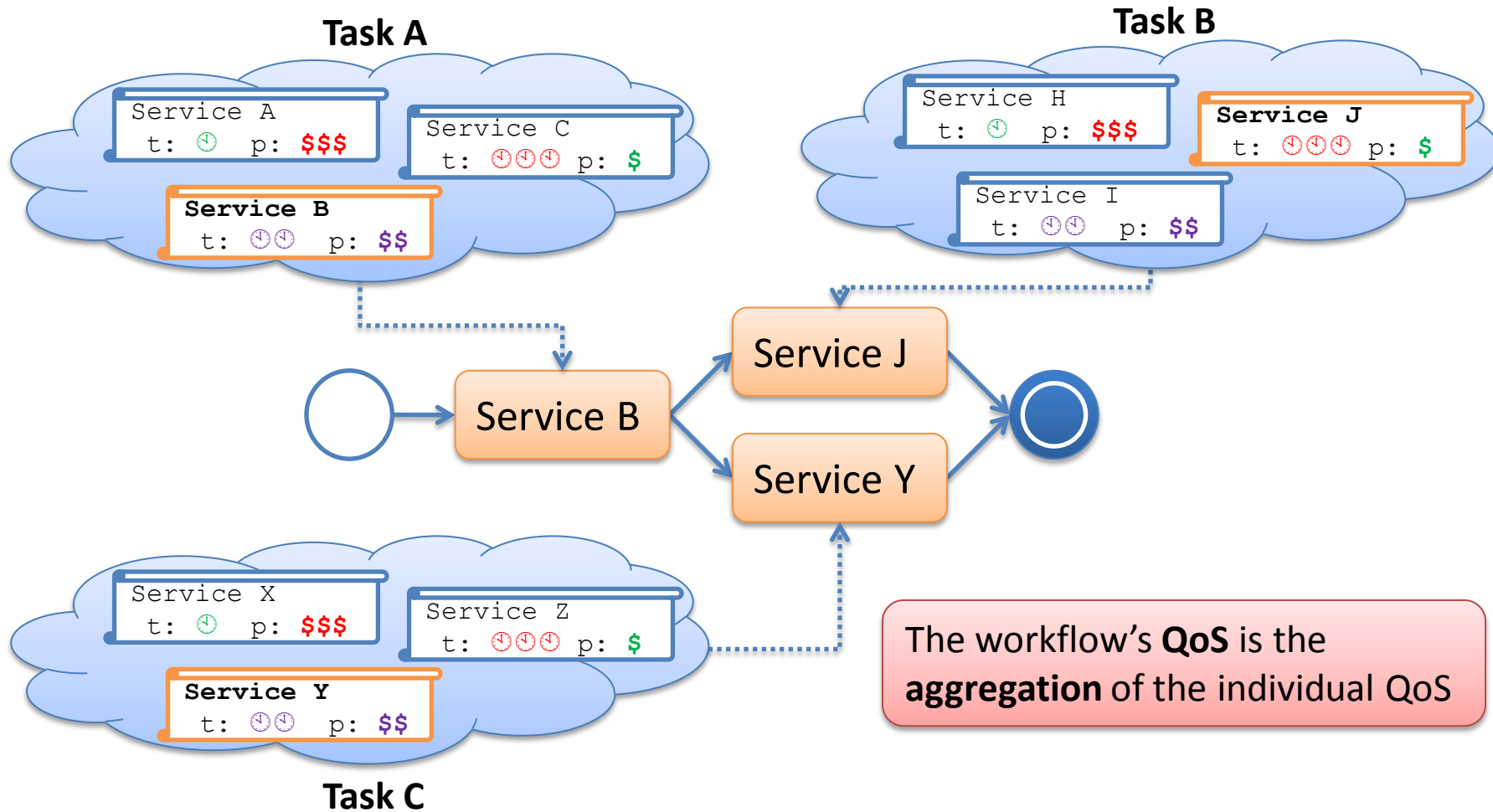


# Quality of Service (QoS):

Assume many **functionally equivalent** services,  
that differ **only** in their **QoS**.

# Quality of Service (QoS)

t := time  
p := price



# Optimization:

Find the **concrete** services  
with the “**best**” QoS.

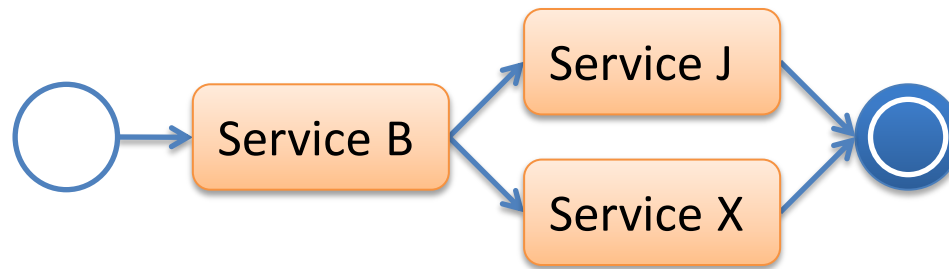
# Optimization Problem

**Minimize:**

weighted sum of QoS

**Fulfill:**

constraints on QoS



E.g: **MIN** { 0.8 **t** + 0.2 **p** }

E.g: ( **t** ≤ 50ms **AND** **p** ≤ 5\$ )

# Challenge:

Finding the **optimal** solution is **NP-hard**.

What are our **CONTRIBUTIONS**



to this **problem...**

## Observation:

Workflows are executed **many times**,  
but we optimize just **a single execution**.

# Standard (Service) Selection Policy

## Services

Service	t	p
<b>Service A</b>	20ms	4\$
<b>Service B</b>	50ms	1\$



## Constraints

### original:

$\text{average}(t) = 40\text{ms}$   
 $\text{monthly}(p) = 240\$$   
 $\text{requests/month} = 120$

### inferred:

$\Rightarrow \text{average}(p) = 2\$$

always A



always B



## QoS over one month

#	S	t	p
1	A	20ms	4\$
2	A	20ms	4\$
...			
120	A	20ms	4\$

$\text{average}(t) = 20\text{ms}$   
 $\text{monthly}(p) = 480\$$

#	S	t	p
1	B	50ms	1\$
2	B	50ms	1\$
...			
120	B	50ms	1\$

$\text{average}(t) = 50\text{ms}$   
 $\text{monthly}(p) = 120\$$

## Idea:

Choose **different** services for each execution **probabilistically**.

# Probabilistic (Service) Selection Policy

## Services

Service	t	p
Service A	20ms	4\$
Service B	50ms	1\$

A with 1/3  
B with 2/3



## Constraints



### original:

average (t) = 40ms  
monthly (p) = 240\$  
requests/month = 120

### inferred:

=> average (p) = 2\$

## QoS over one month

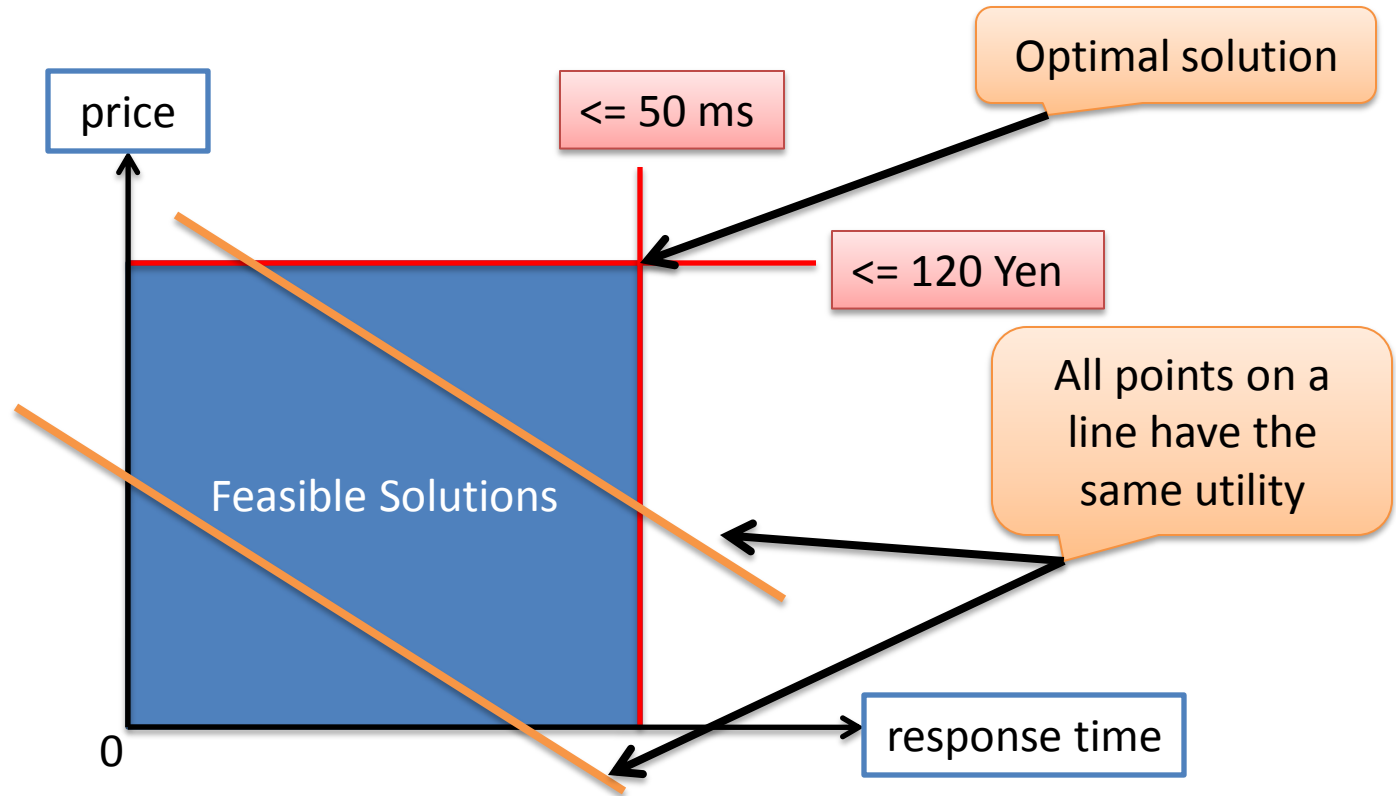
#	S	t	p
1	A	20ms	4\$
2	B	50ms	1\$
3	B	50ms	1\$
...			
118	A	20ms	4\$
119	B	50ms	1\$
120	B	50ms	1\$

average (t) = 40ms  
monthly (p) = 240\$

# Realization:

Use **Linear Programming**  
to obtain an **optimal** solution.

# Linear Programming: Idea



# Benefits:

**Utility** (long-term)

**Performance** (polynomial)

# How does the **EVALUATION**



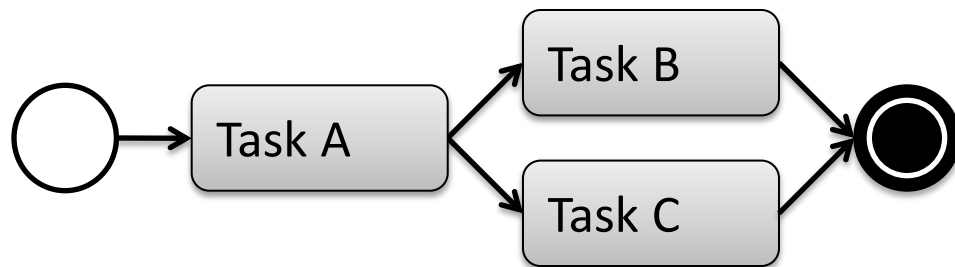
prove our points...

# Performance:

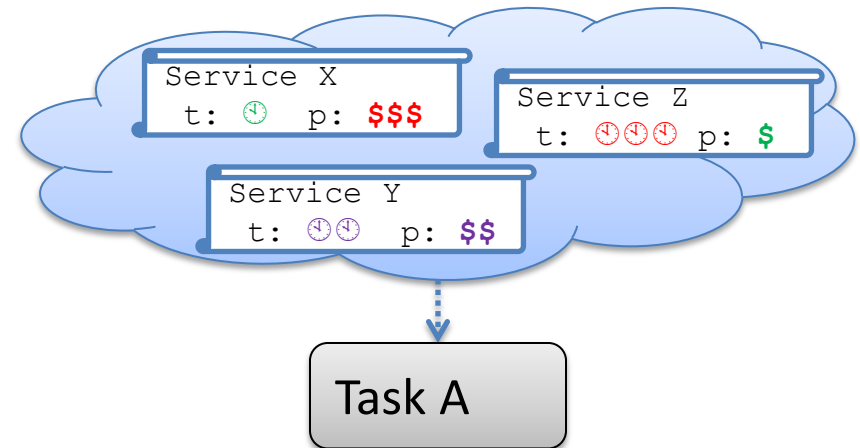
Scaling in regards to  
**#(Services per Task)** and  
**#(Services per Workflow)**

# Key Indicators

Number of **Services per Workflow**



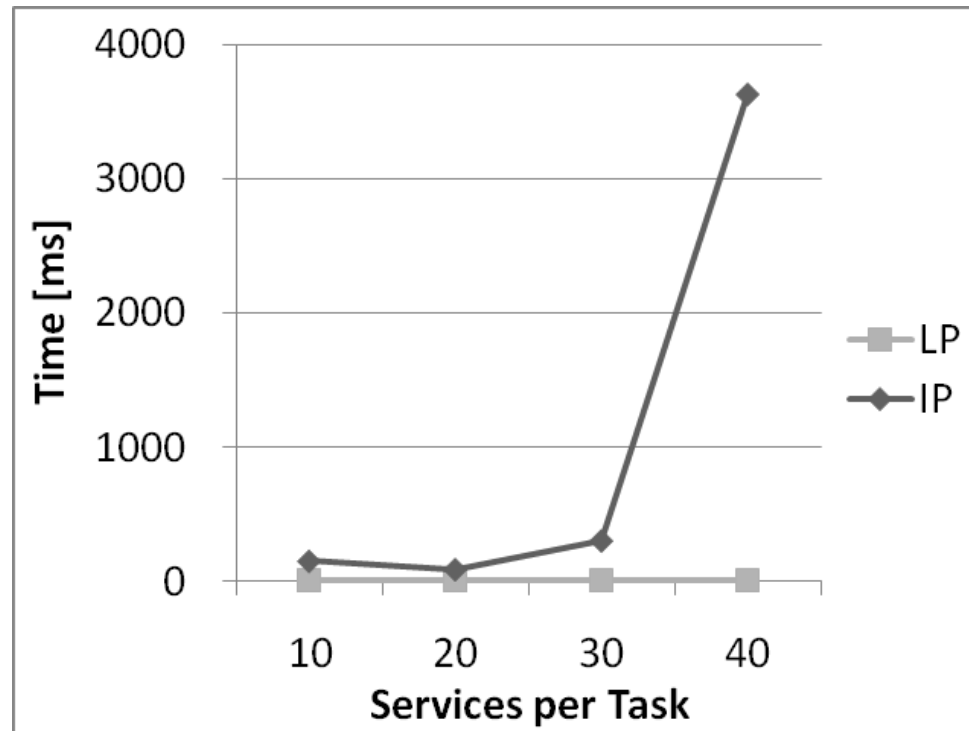
Number of **Services (Candidates) per Task**



# Performance:

## Scalability vs. standard approach

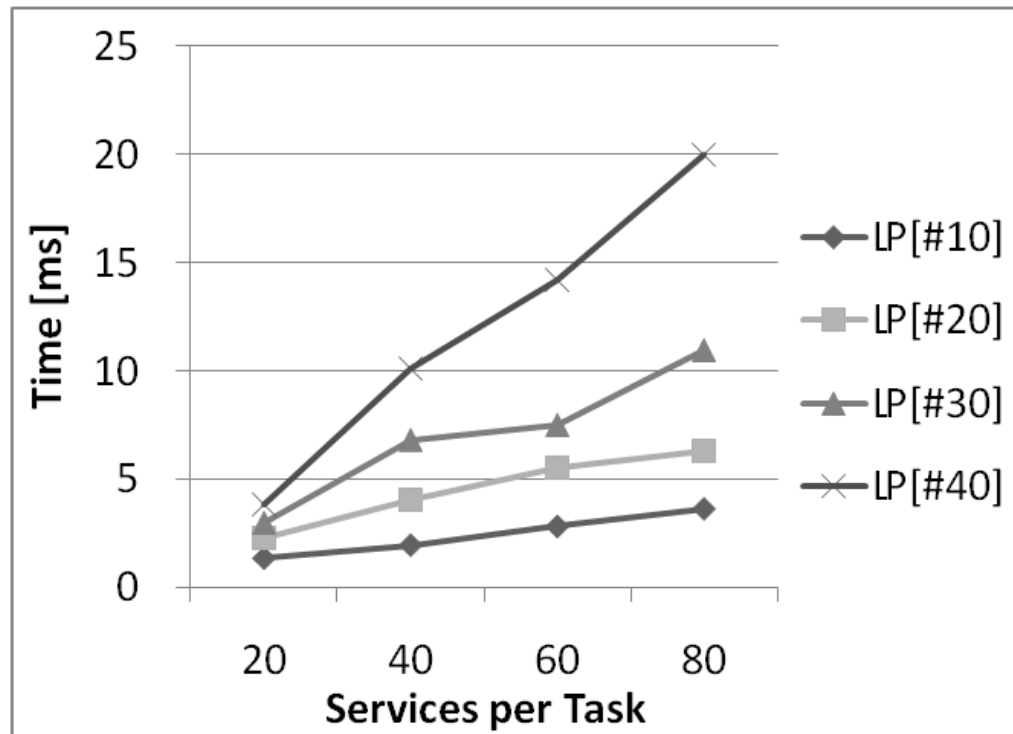
# Linear Programming (LP) vs. Integer Programming (IP)



# Performance:

## Scalability in general

# Scalability



} # (Services per Workflow)

# Utility:

**How much more vs. standard approach...**

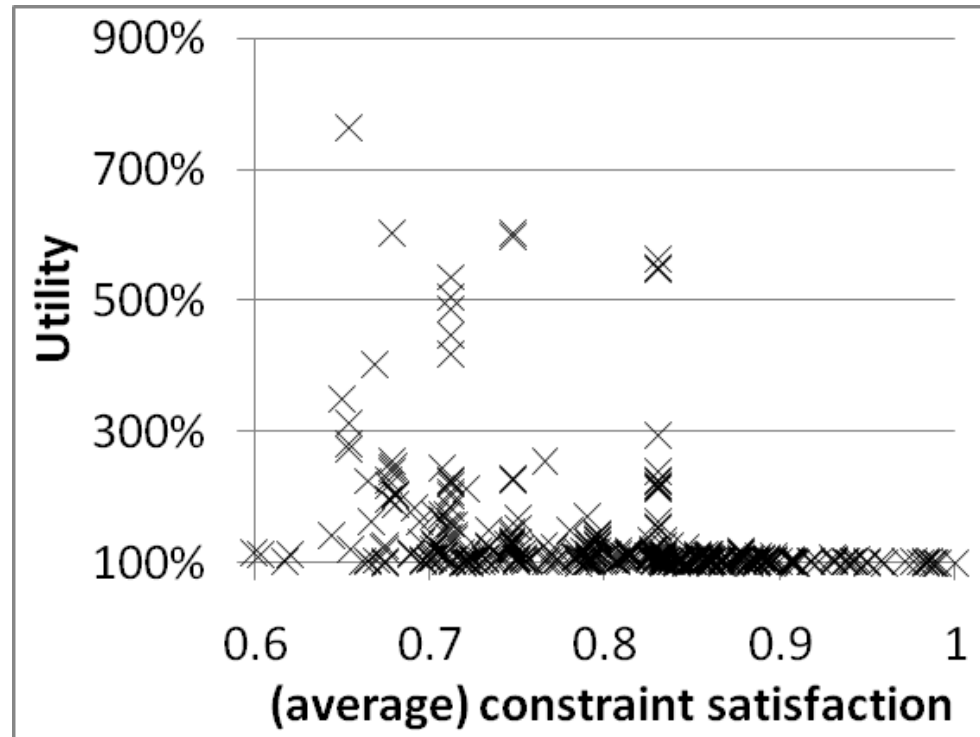
# Comparison

	Standard	Probabilistic
<b>Horizon</b> (Utility, Constraints)	<b>short-term</b> (single execution)	<b>long-term</b> (multiple executions)
<b>Approach</b>	<b>Integer Programming</b> (original problem)	<b>Linear Programming</b> (relaxed problem)
<b>Short-term Constraints</b>	<b>Yes</b> (guarantee 100%)	<b>No</b> (only >60%)
<b>Long-term Constraints</b>	<b>Yes</b> (by short-term ~)	<b>Yes</b> (directly)

- ✓ Monthly budget
- ✓ Availability
- ✓ Average response time
- ✓ Etc.

# Utility

(% of standard approach)

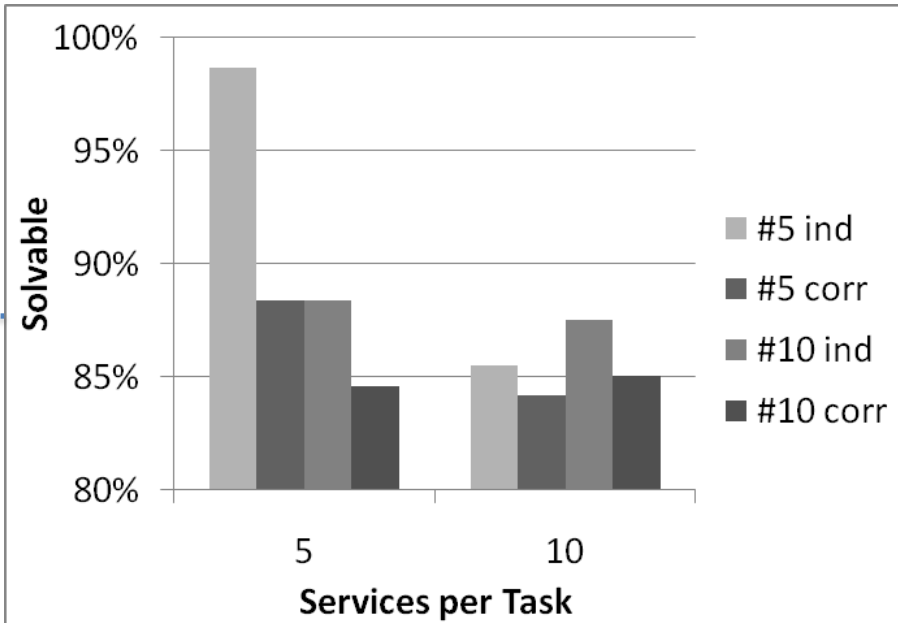


(if constraints are interpreted for **short-term**)

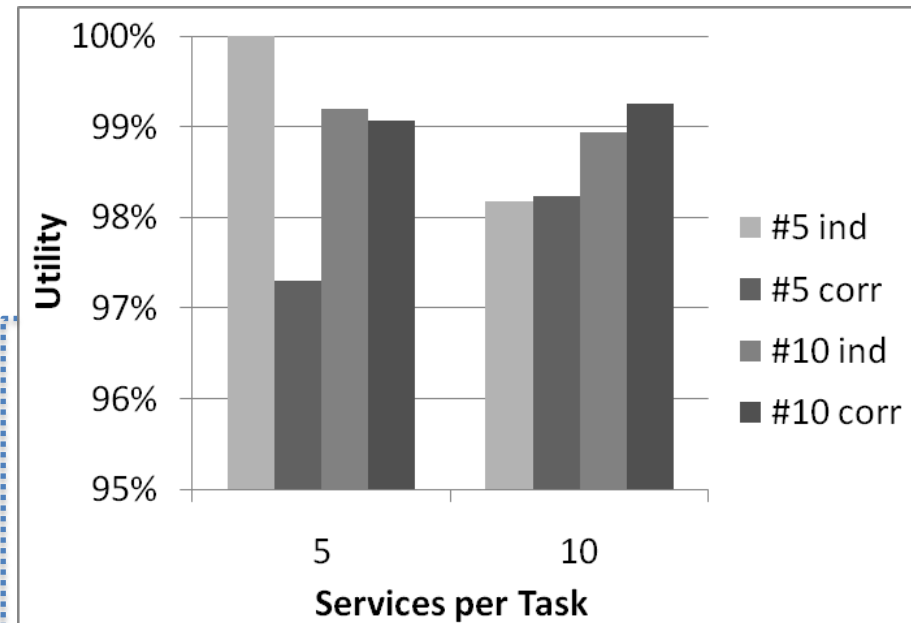
# Beyond:

Can we still tackle the **original problem...**

# Tackle Original IP Problem



→ (choose subset of probabilistic policy)



→ (% of standard approach)

# CONCLUSION:

**Efficient approach optimizing QoS in the long-term**  
**Utility gains for QoS in the long-term**



# FUTURE WORK:

**Potential for solving short-term QoS as well**



東京大学  
THE UNIVERSITY OF TOKYO

NII 大学共同利用機関法人 情報・システム研究機構  
国立情報学研究所  
National Institute of Informatics

Thank you for your attention!

# References

- Pictures

- <http://en.wikipedia.org/wiki/File:Edomaenigiri.jpg>
- [http://en.wikipedia.org/wiki/File:Large Cayenne.jpg](http://en.wikipedia.org/wiki/File:Large_Cayenne.jpg)
- [http://en.wikipedia.org/wiki/File:Skyscrapers of Shinjuku 2009 January.jpg](http://en.wikipedia.org/wiki/File:Skyscrapers_of_Shinjuku_2009_January.jpg)
- [http://en.wikipedia.org/wiki/File:Salmon sushi.jpg](http://en.wikipedia.org/wiki/File:Salmon_sushi.jpg)
- [http://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Hiroshima\\_yaki\\_by\\_woinary at the food court in Hiroshima Airport.jpg](http://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Hiroshima_yaki_by_woinary_at_the_food_court_in_Hiroshima_Airport.jpg)
- [http://en.wikipedia.org/wiki/File:Baked cheesecake with raspberries and blueberries.jpg](http://en.wikipedia.org/wiki/File:Baked_cheesecake_with_raspberries_and_blueberries.jpg)