Optimal service selection policies for dynamic service composition

Miroslav Živković
University of Amsterdam (UvA)
System and Network Engineering Group
m.zivkovic@uva.nl

Joost Bosman, Hans van den Berg, Rob van der Mei, Erik Meeuwissen, Rudesindo Núñez-Queija
• M. Živković, J. Bosman, H. van den Berg, R. van der Mei, H. Meeuwissen, R. Núñez-Queija: Run-time Revenue Maximization for Composite Web Services with Response Time Commitments (AINA 2012)

• M. Živković, H. van den Berg: Revenue Optimization of Service Compositions using Conditional Request Retries (IJWSR 2013)
Service Composition problem
Given abstract workflow select concrete services to execute it

Services implementing task 1

- Service A
  - t: [time icon]
  - p: [price icon]

- Service B
  - t: [time icon]
  - p: [price icon]

Services implementing task 2

- Service K
  - t: [time icon]
  - p: [price icon]

Services implementing task 3

- Service X
  - t: [time icon]
  - p: [price icon]

- Service Y
  - t: [time icon]
  - p: [price icon]

Services implementing task 4

- Service α
  - t: [time icon]
  - p: [price icon]

- Service β
  - t: [time icon]
  - p: [price icon]
Service composition

• We focus on orchestration

• At design time (multi-objective problem)
  – Choices made upfront; non-dominated, Pareto-optimal solutions
  – Inflexible; impossible to modify on-the-fly

• At execution time:
  – Composition choices are made on-the-fly, flexible
What if performance worsens?

- Runtime service selection
- Runtime service substitution
Runtime service selection: Model

Runtime service selection
• Sequential workflow
• Composition may be adapted during execution (per request/task)
• Use of elapsed time info
The orchestrator

- **knows the workflow**
- selects appropriate services
- makes appropriate Service Level Agreements (SLAs) with 3\textsuperscript{rd} party providers and its clients
- has **no impact** to or control of 3\textsuperscript{rd} party domains
- **End-to-end SLA**
  - response time deadline \((\delta_p)\)
  - Reward when response time \(\leq \delta_p\), penalty otherwise
- **Single service SLA**
  - Execution cost, response time distribution
Optimized dynamic decisions

- **Given**
  - Position in workflow
  - Remaining time until deadline
  - Response time distributions
  - Costs, reward and penalty

- **Decision**
  - what service alternative to select based on the elapsed time?

- **Goal**
  - Optimize expected revenue

- **Solution** Apply Dynamic Programming
• **Simple** solution
• Calculate lookup table **off-line**
• Apply lookup table **on-line** (no computing)
Example workflow, 4 tasks
Issue 1: sequential workflow
Issue 2 - availability

request 1 composition: CS1(2)-CS2(2)-CS3(1)-CS4(1)

request 2 composition

response 1

response 2
Runtime service substitution: Model

- For an orchestrated service, at each decision point (A, B, C, D)
  - a) **which service** should be selected?
  - b) **when** is it optimal to perform substitution
  - c) **which service** should be selected for a retry, same or some other?
- with the goal to maximize **end-to-end** expected revenue for given deadline
Response-time: when does a substitution make sense?

- Heavy-tailed response time distributions
  - **Expectation paradox**: “the longer we have waited, the longer we should expect to wait”
- Bimodal/Multimodal distributions
- Substitute by any given service
  - This **does not involve** the costs
Optimal solution

- Use **dynamic programming** to calculate the **policy**:  
  - Compare the expected revenues with and without retry  
  - Formulae for case when **single retry per each task** is allowed  
  - Task $i$, service $j$, deadline $\delta$, retry moment $\theta$, response time distribution $f, F$, revenue $W$

\[
\mathbb{E}[W_j^i(\delta)] = -c_j^i + \int_0^\delta f_j^i(\tau)W^{(i+1)}(\delta - \tau)d\tau + (1 - F_j^i(\delta)) \cdot W^{(i+1)}(0).
\]

\[
\mathbb{E}[W_{j\rightarrow k}^i(\delta, \theta_{j\rightarrow k}^i)] = -c_j^i + \int_0^{\theta_j^i \rightarrow k} f_j^i(\tau)W^{(i+1)}(\theta_j^i \rightarrow k - \tau)d\tau
\]

- **term 1**: execution cost; **term 2** – no retry needed; **term 3** – retry made
Runtime service substitution - conclusions

• For larger values of deadlines, policies with or without retries are close

• Perform substitution for the last tasks
  – Cost plays a role: the more you pay the less substitutions you should perform
Issue 3 – time invariant PDF: Closed loop control

For each alternative response time distribution: keep last n samples
- Calculate empirical distribution(s)
- Apply DP on empirical distributions

Challenges
1. We do not prefer updating the policy after each realization
2. When a certain alternative is never selected we don’t observe changes

Solutions
1. Apply statistical test to see whether an empirical distribution has changed significantly
2. If a service is not used for (specified) time interval send a probe request (and pay corresponding cost);
   - Tradeoff: short interval (good + expensive) vs. large interval
Thank you!