QoS aware service composition in the Internet of Services

Joost Bosman

Agenda:
1. Dynamic runtime (re)composition
2. Closed loop control
3. Demo
4. Experimental results
Orchestrator/Composite Service Provider (CSP)

The CSP

- knows the workflow (could be very complex)
- selects appropriate services
- makes appropriate Service Level Agreements (SLAs) with 3rd party providers and its clients
- has no impact to or control of third party domains
- knows the services’ response times
Model Description

Fully dynamic runtime re-composition:
• Composition may be adapted during execution
• Use of elapsed time info
• For each task, there may be alternatives
Composite SLA

\begin{align*}
S_1,1 & \rightarrow S_1,2 & \rightarrow S_1,3 \\
S_2,1 & \rightarrow S_2,2 & \rightarrow S_2,3 & \rightarrow S_2,4 \\
S_3,1 & \rightarrow S_4,1 & \rightarrow S_4,2
\end{align*}

\textbf{cSLA (end-to-end)}

- \( \delta_p \) e2e RT \textbf{deadline}
- \( R \) \textbf{reward} for response time \( \leq \delta_p \)
- \( V \) \textbf{penalty} for response time \( > \delta_p \)
End-to-end Response Time Deadlines

- \( \leq \) deadline \( \rightarrow \) reward \( R \)

- \( > \) deadline \( \rightarrow \) penalty \( V \)

\( \delta_p \) e2e RT deadline

2\textsuperscript{th} International Summer School, Opatija 27-05-2016
Internal (third party) SLA

\[ D_{i,j} \):

iSLA

- RT of service \( S_{i,j} \) is represented by random variable \( D_{i,j} \)
- \( f_{D_{i,j}} \) response time (RT) PDF of service \( S_{i,j} \)
- \( c_{i,j} \) cost of execution
Dynamic Re-Composition Decisions

Decision: what service alternative to make at task $i$, based on the elapsed response time?

- $\delta_p$: e2e RT deadline

Elapsed time | Response time budget $B$ for $i = 2$
Optimized Dynamic Decisions

- **Given:**
  - position in workflow $i$
  - Time until deadline violated $B$
  - Response time distributions $f_{D_{i,j}}$
  - Costs $c_{i,j}$, Reward $R$ and Penalty $V$

- **Problem:**
  - Optimize expected profit $E[R]$

- **Solution:**
  - Apply Dynamic Programming
  - Backward recursion (Bellman equation)
Challenge: state space

• Continuous response times -> continuum of decision points
• Discretize response times according Choudhury and Houck
Choudhury and Houck

- **Response time random variable** $D$
- **CDF:** $F_D(t) = P(D \leq t)$
- **Discretize RT distributions with stepsize** $h$, **index** $k \geq 0$:

$$q^D_k = \begin{cases} 
F_D(h[k + \frac{1}{2}]) - F_D(h[k - \frac{1}{2}]) & k < T \\
F_D(h[k - \frac{1}{2}]) & k \geq T 
\end{cases}$$

2nd International Summer School, Opatija 27-05-2016
Sum of random variables (Throwing dices)

Any side has probability of 1/6

\[ \begin{array}{c}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 \\
3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 10 \\
5 & 6 & 7 & 8 & 9 & 10 & 11 \\
6 & 7 & 8 & 9 & 10 & 11 & 12
\end{array} \]

There are three ways to throw 4: 1+3, 2+2, 3+1
There are six ways to throw 7: 1+6, 2+4, 3+4, 4+3, 5+2, 6+1
Sum of random variables
(General case)

Sum of random variables -> Numerical convolution
• Take all combinations that will add up to k:

\[ D_1 \ast D_2 \rightarrow q^{D_1 \ast D_2}_k = \sum_{j=0}^{k} q^{D_1}_j q^{D_2}_{k-j} \]
The Bellman equation
\textit{(dynamic programming equation)}

- Expected revenue $P_b^{(i)}$, reward $R_b^{(i,j)}$ and penalty $V_b^{(i,j)}$
  - $b$ time-units left until deadline violation
  - $i$ position in the workflow
  - $j$ alternative at position $i$ in the workflow

- Backwards recursion
- Start with final revenue:
- Use optimal expected revenue as input one step back in the workflow
The Bellman equation

\[ P_b = \begin{cases} R & \text{if } b > 0 \\ -V & \text{otherwise} \end{cases} \]
The Bellman equation

- $R_b^{(4,j)} = \sum_{k=0}^{b} q_k^{D_{4,j}} P_{b-k}^{(5)}$ (Expected reward for $j$)
- $V_b^{(4,j)} = \sum_{k=b+1}^{T} q_k^{D_{4,j}} P_0^{(5)}$ (Expected penalty for $j$)
- $P_b^{(4)} = \max \left\{ -c_{4,1} + R_b^{(4,1)} + V_b^{(4,1)}, -c_{4,2} + R_b^{(4,2)} + V_b^{(4,2)} \right\}$
The Bellman equation

- $R_b^{(3,j)} = \sum_{k=0}^{b} q_k^{D_{3,j}} P_b^{(4)} - k$  
  (Expected reward for $j$)
- $V_b^{(3,j)} = \sum_{k=b+1}^{T} q_k^{D_{3,j}} P_0^{(4)}$  
  (Expected penalty for $j$)
- $P_b^{(3)} = \text{max}\left\{-c_{3,1} + R_b^{(3,1)} + V_b^{(3,1)}\right\}$

Proceed one step back
The Bellman equation

- $R_b^{(2,j)} = \sum_{k=0}^{b} q_k^{D_{2,j}} P_b^{(3)} - k$ (Expected reward for $j$)
- $V_b^{(2,j)} = \sum_{k=b+1}^{T} q_k^{D_{2,j}} P_0^{(3)}$ (Expected penalty for $j$)
- $P_b^{(2)} = \max \left\{ -c_{2,1} + R_b^{(2,1)} + V_b^{(2,1)}, \ldots, -c_{2,4} + R_b^{(2,4)} + V_b^{(2,4)} \right\}$

Proceed one step back
The Bellman equation

- \( R_b^{(1,j)} = \sum_{k=0}^{b} q_k^{D_{1,j}} \Pi_{b-k} \)  
  (Expected reward for \( j \))

- \( V_b^{(1,j)} = \sum_{k=b+1}^{T} q_k^{D_{1,j}} \Pi_{0} \)  
  (Expected penalty for \( j \))

- \( P_b^{(1)} = \max \left\{ -c_{1,1} + R_b^{(1,1)} + V_b^{(1,1)} , -c_{1,2} + R_b^{(1,2)} + V_b^{(1,2)} , -c_{1,3} + R_b^{(1,3)} + V_b^{(1,3)} \right\} \)

Until the starting point

2th International Summer School, Opatija 27-05-2016
\( P_{\delta_p} \) Is the expected revenue for deadline \( \delta_p \)

For each step the optimized decisions \( A_b^{(i)} \) can be found by:

\[
A_b^{(i)} = \text{argmax} \left\{ -c_{i,1} + R_b^{(i,1)} + V_b^{(i,1)} \right\}
\]

\[
\vdots
\]

\[
-c_{i,M} + R_b^{(i,M)} + V_b^{(i,M)}
\]

- Position \( i \)
- Alternative \( j \)
- \( M \) alternatives at \( i \)
Dynamic program

- For each service $S_{i,j}$:
  - RT probability: $q_k^{D_{i,j}}$ (for $k$ time units)
  - Invocation cost $c_{i,j}$
- Remaining budget $b$, workflow position $i$

Recursive dependency in $i$

- $R_{b}^{(i,j)} = \sum_{k=0}^{b} q_k^{D_{i,j}} p_{b-k}^{(i+1)}$ (Expected reward for $j$)
- $V_{b}^{(i,j)} = \sum_{k=b+1}^{T} q_k^{D_{i,j}} p_{0}^{(i+1)}$ (Expected penalty for $j$)
- $P_{b}^{(i)} = \max_j \{-c_{i,j} + R_{b}^{(i,j)} + V_{b}^{(i,j)}\}$ (Expected revenue at position $i$)
- $A_{b}^{(i)} = \arg\max_j \{-c_{i,j} + R_{b}^{(i,j)} + V_{b}^{(i,j)}\}$ (Choice with $b$ left till deadline at pos $i$)
- Expected revenue for e2e deadline $\delta_p$: $P_{\delta_p}^{(1)}$
Resulting strategy

\[ A_b^{(i)} = \arg\max_j \left\{ -c_{i,j} + R_{b}^{(i,j)} + V_{b}^{(i,j)} \right\} \]

Example strategy \( A_b^{(i)} \)

Workflow direction

Task 1

Task 2

Task 3

Task 4

Service alternative 4

Service alternative 3

Service alternative 2

Service alternative 1

\[ \text{Time budget } B \rightarrow \]

Overall deadline \( \delta_p \)
Lookup Table

**Approach:**
- Simple solution
- Calculate lookup table off-line
- Apply lookup table on-line (no computing)
- Closed-loop control possible based on monitoring
Agenda

• Up till now:
  – Known response-time distributions

• Next:
  – Decisions based on monitoring/measuring
Empirical distribution
*(based on measurements)*

- **RT random variable** $D$
- **$N$ Measurements** $d_1, d_2, \cdots, d_N$
- **Empirical distribution** $F_D(t) = \frac{1}{N} \sum_{j=1}^{N} 1\{d_j \leq t\}$
- **Discretized empirical distribution (histogram):**

$$q_{k} = \frac{1}{N} \sum_{j=1}^{N} 1\{h(k-0.5) \leq d_j < h(k+0.5)\}$$
Closed-Loop Control

- Updating the strategy with each sample is expensive
- Keep track of reference distribution at each change
- Compare against reference distribution
- When difference to reference is too large
  - Calculate/update empirical distribution(s)
  - Apply backward recursion on empirical distributions
Tracking change

• Sliding window $W$

$$q^\tilde{D}_k = \frac{1}{W} \sum_{j=0}^{W-1} 1\{(k-0.5)\leq d_{t-j} < (k+0.5)\}$$

• Exponential smoothing $\alpha$

$$q^\tilde{D}_k = (1 - \alpha)q^\tilde{D}_{k-1} + \alpha 1\{(k-0.5)\leq d_{t-1} < (k+0.5)\}$$

– No bookkeeping of realizations necessary!

– Virtual sliding window $W = \frac{1+\alpha}{1-\alpha}$
Kolmogorov Smirnov Test

- Imagine we are comparing two empirical PDF’s
- Compute the maximal vertical distance $D_W$
- $D_W$ too large -> significant difference
- Max difference is a pre-computed value $c(\alpha, W)$

$$D_W := \sup_k |q_k - \tilde{q}_k| > c(\alpha, W)$$
Statistical Test

- **Modified Kolmogorov Smirnov test:**
  
  \[ D_W := \sup_k |q_k - \tilde{q}_k| \]

  - Where \( k \) is the discretization index and \( D_W \) is the statistic.

  - For the smoothing approach choose:
    \[ W = \frac{1 + \alpha}{1 - \alpha} \]
Probes

• Certain services are unattractive
  – May not invoked at all

• Probe
  – If service not used in $t_p$ requests then send a dummy request (incurring probe cost)
Closed-loop Control overview

1. **Execute DP**
   - If significant change detected, proceed with next step.
   - If no significant change detected, return to the start.

2. **Update response time info**
   - Use the response time realizations $d_n^{(i,j)}$.

3. **Probe time expired?**
   - If yes, proceed to the next step.
   - If no, go back to the previous step.

4a. **Get DP reference & current info**
4b. **Send probe**
   - If no significant change detected, return to the start.
   - If significant change detected, proceed with next step.

5a. **Test distribution change**
   - If no significant change detected, return to the start.
   - If significant change detected, proceed with next step.

5b. **Update info with probe**
6. **Calculate DP**
7. **Store DP reference info**

---

2th International Summer School, Opatija 27-05-2016
Test bed environment

- Test lab environment developed
- Based on JAVA + NIO
  - Event based, single threaded design
  - Real HTTP traffic
  - Delays generated using SSJ library: (http://simul.iro.umontreal.ca/ssj/indexe.html)
- Algorithms in Matlab using IP control interface
- Trigger mechanism to simulate “Sub service disasters”
- Currently running as multiple threads
- Future: Run on a multiple server test bed
Classical ‘One thread per channel’ model

- Not scalable
- Context switching overhead

Diagram showing:
- ServerSocket for incoming request handling
- Socket for response handling
- Socket for dispatching handling

Thread to Channel flow:
- Thread
- Channel

Diagrams illustrating the flow of handling requests and responses with threads and channels.
Reactor IO model

- Selector handles IO in FIFO fashion

![Diagram of Reactor IO model]

One Thread

Selector

- ServerSocketChannel
  - Incoming Request Handling
- SocketChannel
  - Response Handling
- SocketChannel
  - Dispatching Handling

Thread

Selector

Channel Channel Channel
Test bed setup

Orchestrator (composite service)

- Dispatcher
- Service registry + interface
- Control interface

Real HTTP IP traffic

Service registry + interface

Control interface

Sub Service 1 (atomic service)

Service interface

Control interface

Sub Service N (atomic service)

Service interface

Control interface

Lab load generator

- Control interface

Requestor

Control interface

Matlab

Control interface

Algorithms run in Matlab

HTTP Connection

Experiment Control Connection
Demo

Swap services after 2000 requests
No Control Updates (red line)
Control updates (blue line)
Results

- **Swap service** $(i,2)$ and $(i,3)$ at $i = n_{\text{swap}}$ after 5000 requests
- **Parameters:**
  - $W$ Window size
  - $t_p$ Probe interval
  - $\alpha$ Test significance level
  - $n_{\text{swap}}$ Swap position (experiment parameter)
Window size $W$

$n_{\text{swap}} = 1$

$n_{\text{swap}} = 3$

$W = 25$

$W = 50$

$W = 100$

Swap services event

Recovery from swap event
Position in chain

Swap services event

\[ n_{\text{swap}} = 1 \quad n_{\text{swap}} = 2 \quad n_{\text{swap}} = 3 \quad n_{\text{swap}} = 4 \]
Probe interval $t_p$

Swap services event

$t_p = 10$
$t_p = 25$
$t_p = 50$
$t_p = 100$
Closed-loop Control

• **Challenges:**
  1. It is not desirable to update the policy after each realization.
  2. If a policy never selects a certain alternative 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 \( t_p \) times send a probe request (and pay corresponding cost).
Tradeoffs

Window size for PDF updates:

- Big $W$: good estimate of distribution but slow response to changes
- Small $W$: poor estimate of distribution but quick response to changes

Frequency of probing for PDF updates:

- Small $t_p$: good updates of rarely used alternatives but higher cost
- Big $t_p$: poor updates of rarely used alternatives but lower cost
Work in progress

- Take in account sample age (invocation time)
- Non-linear composition structures

**Conclusion:**
- Dynamic workflow composition extremely powerful
- Many challenging open research questions