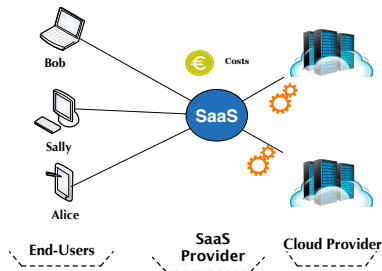


Négociation multi-agent « un-à-plusieurs » et mécanismes de coordination pour la gestion de la satisfaction des utilisateurs d'un service

Amro Najjar Yazan Mualla Kamal Singh Gauthier Picard



Gestion de la satisfaction des utilisateurs

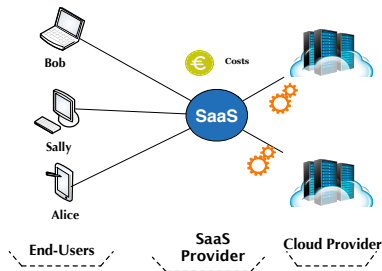


Gestion de la satisfaction (SM)

Le fournisseur de service cherche un équilibre entre :

1. minimiser des coûts et de l'utilisation des ressources
2. fournir un service qui satisfait (à **grain fin**) les utilisateurs du service

Gestion de la satisfaction des utilisateurs



Gestion de la satisfaction (SM)

Le fournisseur de service cherche un équilibre entre :

1. minimiser des coûts et de l'utilisation des ressources
2. fournir un service qui satisfait (à **grain fin**) les utilisateurs du service

Problème des solutions actuelles

- la décision SM is prise unilatéralement par le fournisseur
- les préférences et les attentes de l'utilisateur final sont ignorées
- les différences entre utilisateurs sont ignorées

Proposition

Développer une architecture « un-à-plusieurs » dont les objectifs sont :

- Intégrer l'évaluation subjective de la qualité du service par les utilisateurs
[HOSSFELD et al., 2016; NAJJAR, 2015]
- Proposer au fournisseur un contrôle fin sur la satisfaction des utilisateurs tout en respectant des contraintes budgétaires

Plan

Qualité d'expérience

AQUAMan : Adaptive-QoE-Aware elasticity Management

Gestion fine de la satisfaction dans AQUAMan

Évaluation

Qualité d'expérience (QoE)

Définition (Qualité d'expérience)

La qualité d'expérience (QoE) est la qualité d'un service estimée *subjectivement* par un utilisateur final (mesure subjective et centrée utilisateur)

Qualité d'expérience (QoE)

Définition (Qualité d'expérience)

La qualité d'expérience (QoE) est la qualité d'un service estimée *subjectivement* par un utilisateur final (mesure subjective et centrée utilisateur)

Défi

La plupart des travaux sont *unilatéraux* et repose sur le MOS (Mean Opinion Score), qui masque des informations sur la diversité des utilisateurs et ignore leur évolution

[HOSSFELD et al., 2011]

Qualité d'expérience (QoE)

Définition (Qualité d'expérience)

La qualité d'expérience (QoE) est la qualité d'un service estimée *subjectivement* par un utilisateur final (mesure subjective et centrée utilisateur)

Défi

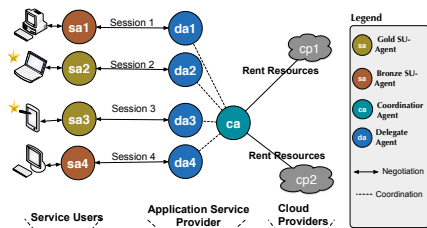
La plupart des travaux sont *unilatéraux* et repose sur le MOS (Mean Opinion Score), qui masque des informations sur la diversité des utilisateurs et ignore leur évolution

[HOSSFELD et al., 2011]

Solution possible

- De récentes études empiriques recommandent aux fournisseurs de se baser sur les **percentiles** pour évaluer la satisfaction des utilisateurs [HOSSFELD et al., 2016]
- Les agents peuvent intégrer les préférences **personnelles et évolutives** des utilisateurs et les impliquer dans le processus de décision [NAJJAR et al., 2016a, 2017]

Architecture d'AQUAMan



- Intègre les utilisateurs dans la boucle [NAJJAR et al., 2016a]
- Permet une gestion de la QoE et personnelle de la gestion de l'élasticité du service [NAJJAR et al., 2016b]
- Les utilisateurs ont des fonctions d'utilité personnelles et des stratégies de négociation propres
- Les sessions de négociation sont autonomes mais doivent respecter des contraintes budgétaires RC
- Les utilisateurs peuvent entrer ou quitter le système à volonté

Gestion de la satisfaction dans AQUAMan

Pendant le processus de négociation, les utilisateurs peuvent accepter ou rejeter un service en fonction de leur attentes et de l'évaluation subjective de sa qualité

Mécanisme d'adaptation

- **But** : garantir des objectifs prédéfinis de satisfaction sans dépasser un budget RC , le coût moyen investi par utilisateur
- **Solution** : le fournisseur doit
 1. construire un modèle de comportement de négociation pour chaque utilisateur
[BAARSLAG et al., 2015; NAJJAR et al., 2017]
 2. ajuster sa stratégie de négociation afin de restaurer la satisfaction des utilisateurs au niveau désiré, tout en respectant RC

Modélisation

Objectifs de satisfaction de utilisateurs

- Imposer $Goal[k]$, i.e. quel pourcentage d'utilisateurs reçoivent un service de catégorie de qualité k (e.g. *acceptable*, *good*, *excellent*)

Gestion de la satisfaction comme un problème d'optimisation

$$\text{minimiser } \sum_{k=1}^K \sum_{i=1}^{N^t} C_i^k \cdot X_i^k \quad (1)$$

avec

$$\sum_{k=1}^K X_i^k \leq 1, \quad \forall i \quad (2)$$

$$\sum_{m=1}^k \left(\frac{\sum_{i=1}^{N^t} X_i^m + Already[m]}{\#TerminatedSessions} \right) \geq Goal[k], \quad \forall k \quad (3)$$

Algorithme heuristique

Construire une liste des utilisateurs approchant leur date d'échéance

Tri de la liste, trois stratégies

1. $\overline{c_i}$: les utilisateurs avec des exigences d'*acceptabilité* moins chères sont privilégiés
2. $\overline{u_{sa_i}}$: les utilisateurs avec la *satisfaction* la plus faibles sont privilégiés
3. $\overline{T_{sa_i}}$: les utilisateurs pour lesquels l'échéance est proche sont privilégiés

Sélectionner les utilisateurs dans la liste, deux stratégies

1. *DISTANCE_TO_GOAL* : une catégorie k , dont la distance pour atteindre son objectif $Goal[k]$ est plus grande, a la priorité sur les autres catégories
2. *CATEGORY_COST* plus une catégorie est satisfaisante, plus elle a priorité sur les autres

Sur la base du modèle utilisateur, proposer une offre sur mesure

Solveur optimal (CPLEX) vs. algorithme heuristique

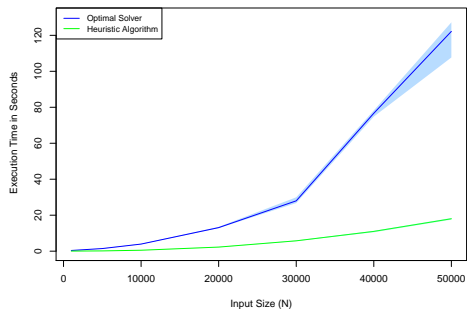


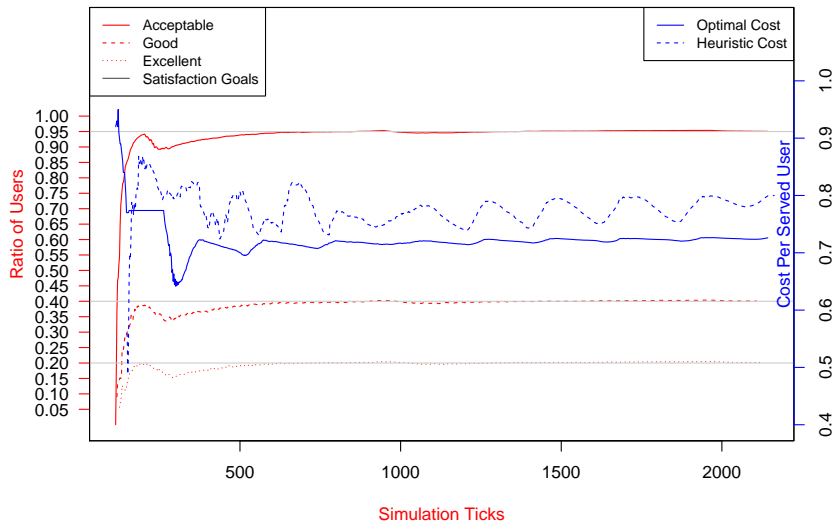
TABLE – Surcoûts de l'approche heuristique (vs. coûts optimaux)

	Utility_First	Cost_First	Deadline_First
DISTANCE_TO_GOAL	+10.2%	+16.5%	+10.9%
CATEGORY_COST	+12.9%	+12.9%	+13.1%

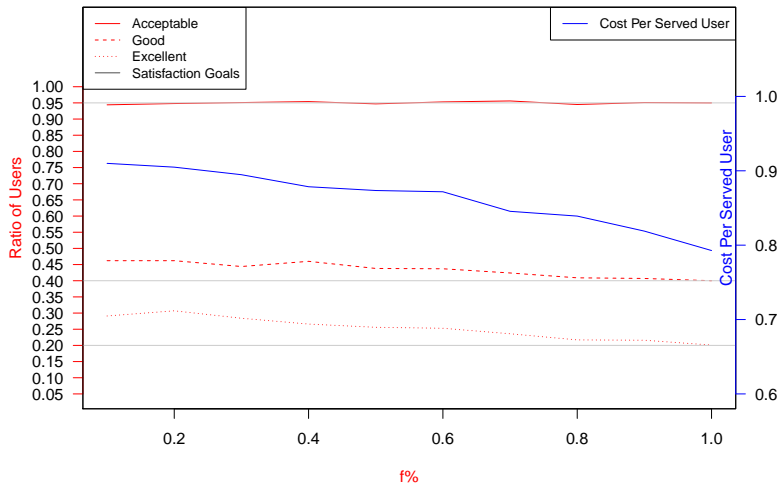
Résultats

- Le solveur optimal ne passe pas à l'échelle avec un nombre d'utilisateurs grandissant
- Concernant les coûts, la meilleure heuristique est seulement 10% plus chère que le coût optimal

Satisfaction des utilisateurs & coût



Impact du retour de utilisateurs



Conclusions & perspectives

Conclusions

- AQUAMan* fournit à l'utilisateur un contrôle fin sur sa satisfaction
- AQUAMan* satisfait les contraintes budgétaire du fournisseur
- AQUAMan* est réactif et utilisable en temps réel

Conclusions & perspectives

Conclusions

- AQUAMan* fournit à l'utilisateur un contrôle fin sur sa satisfaction
- AQUAMan* satisfait les contraintes budgétaire du fournisseur
- AQUAMan* est réactif et utilisable en temps réel

Travaux futurs

- Capturer les politiques de partage de préférences
- Apprendre plus finement le comportement des utilisateurs (BBC)

Négociation multi-agent « un-à-plusieurs » et mécanismes de coordination pour la gestion de la satisfaction des utilisateurs d'un service

Amro Najjar Yazan Mualla Kamal Singh Gauthier Picard

Références



BAARSLAG, Tim, Mark JC HENDRIKX, Koen V HINDRIKS et Catholijn M JONKER (2015). “Learning about the opponent in automated bilateral negotiation : a comprehensive survey of opponent modeling techniques”. In : *Autonomous Agents and Multi-Agent Systems*, p. 1–50.



HOSSFELD, Tobias, Raimund SCHATZ et Sebastian EGGER (2011). “SOS : The MOS is not enough!”. In : *QoMEX*, p. 131–136.



HOSSFELD, Tobias, Poul E HEEGAARD, Martín VARELA et Sebastian MÖLLER (2016). “Qoe beyond the mos : an in-depth look at qoe via better metrics and their relation to mos”. In : *Quality and User Experience 1.1*.



NAJJAR, Amro (2015). “Multi-Agent Negotiation for QoE-Aware Cloud Elasticity Management”. Thèse de doct. École nationale supérieure des mines de Saint-Étienne.



NAJJAR, Amro, Christophe GRAVIER, Xavier SERPAGGI et Olivier BOISSIER (2016a). “Modeling User Expectations Satisfaction for SaaS Applications Using Multi-agent Negotiation”. In : *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, p. 399–406. DOI : 10.1109/WI.2016.0062.



NAJJAR, Amro, Xavier SERPAGGI, Christophe GRAVIER et Olivier BOISSIER (2016b). “Multi-agent Systems for Personalized QoE-Management”. In : *Teletraffic Congress (ITC 28), 2016 28th International*. T. 3. IEEE, p. 1–6.



NAJJAR, Amro, Yazan MUALLA, Olivier BOISSIER et Gauthier PICARD (2017). “AQUAMan : QoE-Driven Cost-Aware Mechanism for SaaS Acceptability Rate Adaptation”. In : *2017 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*.

Impact of BBC Users

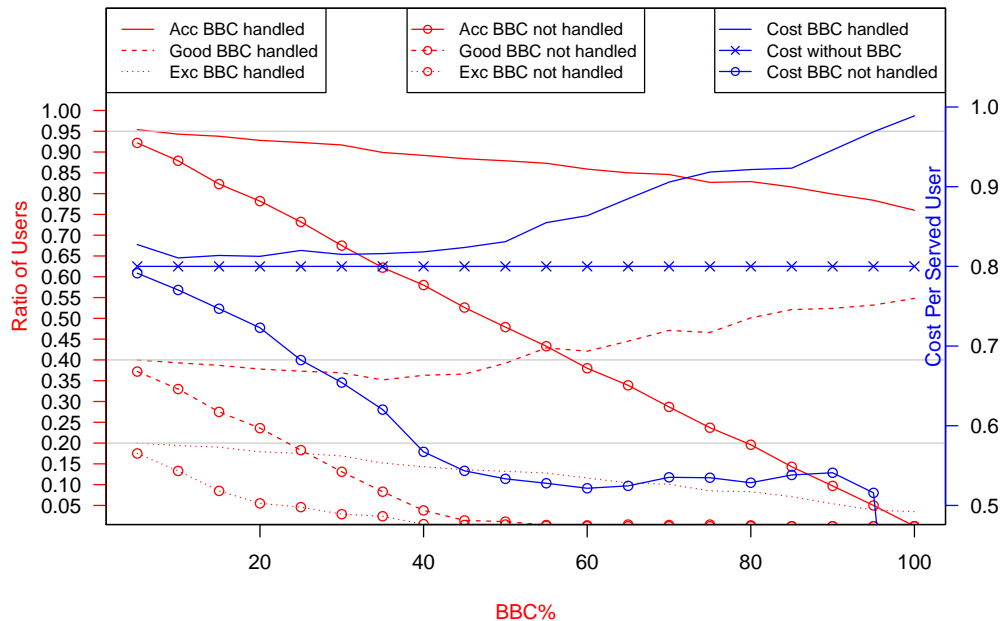
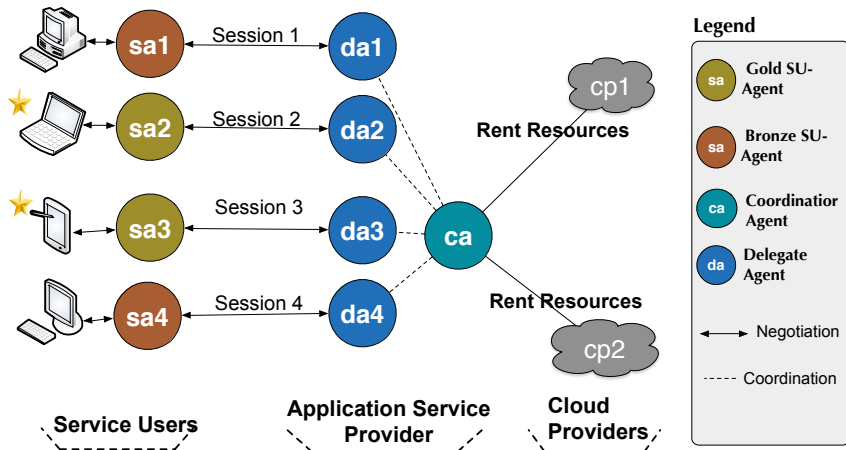


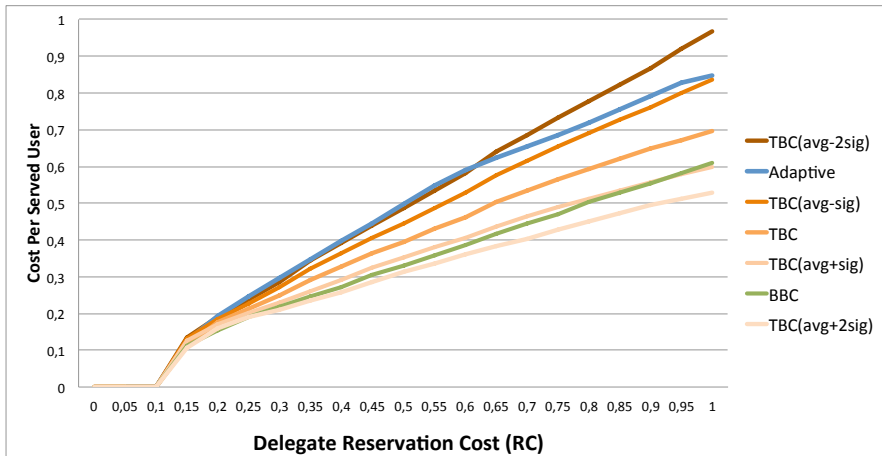
FIGURE – The Impact of BBC Users.

Limit of the Adaptive Mechanism



- We set $Goal = 100\%$ to push AQUAMan to accept as much user as possible in order to identify its limit

Costs of the Adaptive Mechanism



The cost spent per user of AQUAMan compared with other non-adaptive strategies

Elasticity Negotiation : Negotiation Object

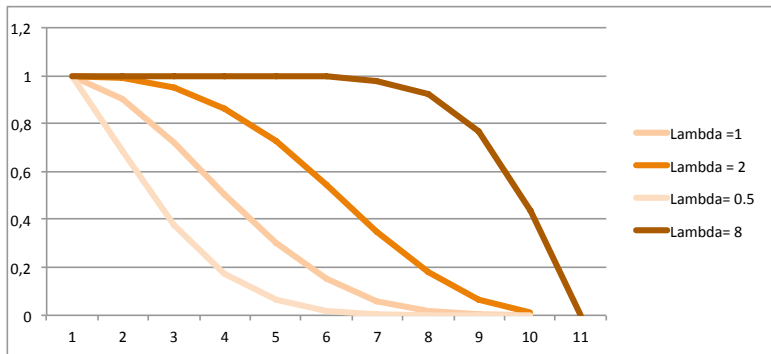
Offer

- The negotiation object is offer $o_i^t = \langle v_1, v_2, \dots v_J \rangle$
- It assigns a value v_k ($k \in [1, J]$) to each of the service attribute

Example : in the transcoding service an offer is :

$$o_{da_i}^t = \langle HD, 10 \text{ min}, 24p \rangle$$

Concession Strategy



- sa_i make concessions by reducing their aspiration rate

- $\Delta AR_i^t = AR_i^t \cdot \left(\frac{t}{T_{sa_i}}\right)^{\lambda_i}$

- Then $AR_i^{t+1} = AR_i^t - \Delta AR_i^t$

- When T_{sa_i} is reached, sa_i quits the negotiation

λ controls the convexity of the concession curve

- $\lambda_i < 1$: Conciliatory
- $\lambda_i \approx 1$: linear
- $\lambda_i > 1$: Boulware

Linear User Utility functions

- sa_i rely on a utility function M derived from su_i preferences
- M is a weighted sum of μ_{sa_i} , linear attribute utility functions

$$\mu_{sa_i, at_j}(o_{da_i}^t[at_j]) = \frac{rv_{sa_i, at_j} - o_{da_i}^t[at_j]}{rv_{sa_i, at_j} - pv_{sa_i, at_j}} \quad (4)$$

- rv_{sa_i, at_j} and pv_{sa_i, at_j} are the preferred and reservation values of the attribute at_j
- sa_i 's subsequent decision is based on the acceptance condition

QoE Influence Factors

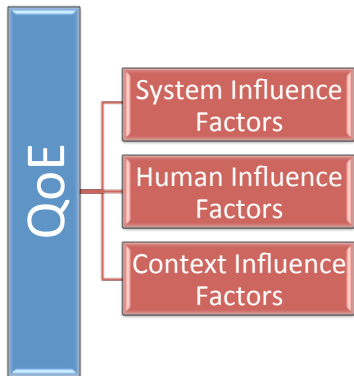
QoE is a function of influence factors. Influence factors are the independent variables and QoE is dependent variable :

$$QoE_s = \Phi(IF_1, IF_2, \dots, IF_n)$$

QoE Influence Factors

QoE is a function of influence factors. Influence factors are the independent variables and QoE is dependent variable :

$$QoE_s = \Phi(IF_1, IF_2, \dots, IF_n)$$



1. **SIF** :technical aspects of the system on which the user consumes the factors
2. **HIF** :user background, expectations, his personality and previous experience
3. **CIF** : the context when the user consumes the service

QoE-aware Agent Decision Model

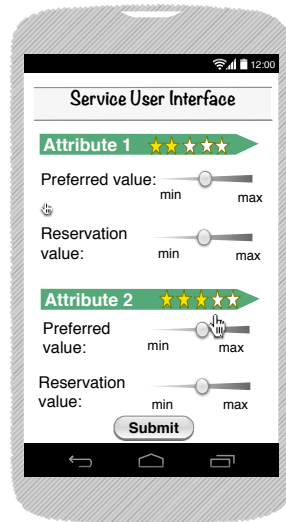
- The user agent seeks to maximize the **subjective satisfaction** of the user
- Therefore, it derives its decision model and utility function from the user preferences
 - ▶ Thus, for each attribute :

$$\mu_{sa_i, at_j} = \alpha_{sa_i, at_j} \cdot f(v_{at_j}) + \beta_{sa_i, at_j} \quad (5)$$

- ▶ The form of $f()$ depends on the attribute type
- ▶ If the service involves multiple attributes :

$$M_{sa_i}(o) = \sum_{j=1}^{j=J} w_{sa_i, j}^t \cdot \mu_{sa_i, at_j}(o_{da_i}^t[at_j]) \quad (6)$$

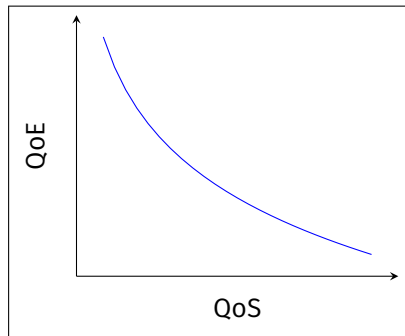
- ▶ Where o is an offer received from the provider



The Logarithmic Hypothesis

- Derived from the Weber Fechner Law (WFL) of psychophysics
- Logarithmic relationship between QoS parameter and QoE :

$$QoE = -\alpha \cdot \log(QoS) + \beta \quad (7)$$



- QoS should be perceivable by the user (e.g. waiting time)
- Validated and applied to various applications (file download, Email, etc.)

QoE-Aware Version : Decision Model

M becomes a weighted sum of **logarithmic** attribute utility functions :

$$M_{sa_i}(o_{da_i}^t) = \sum_{j=1}^{j=J} w_{sa_i,j}^t \cdot [-\alpha_{sa_i,at_j} \cdot \ln(o_{da_i}^t[at_j]) + \beta_{sa_i,at_j}] \quad (8)$$

Coefficients

- $\alpha_{sa_i,at_j}, \beta_{sa_i,at_j}$ are the **personal** coefficients of sa_i for at_j
- Derived from sa_i preferences :

$$\alpha_{sa_i} = \frac{1}{\ln(rv_{i_{res}}) - \ln(pv_{i_{res}})} \beta_{sa_i} = \frac{\ln(rv_{i_{res}})}{\ln(rv_{i_{jct}}) - \ln(pv_{i_{res}})} \quad (9)$$

Use-case scenario : Cloud-hosted Service

Provider (ASP) Side

- A SaaS provider whose service is compute-intensive
- Rents resources from a cloud provider (CP) (e.g. Amazon EC2) to fulfill the requests

Use-case scenario : Cloud-hosted Service

Provider (ASP) Side

- A SaaS provider whose service is compute-intensive
- Rents resources from a cloud provider (CP) (e.g. Amazon EC2) to fulfill the requests

User Side

- We suppose that all attributes of the service are directly perceivable by the user (e.g. execution time)
- Therefore, we consider that the logarithmic hypothesis applies to these attributes

$$\mu_{sa_i, at_j}(v_{at_j}) = -\alpha_{sa_i, at_j} \cdot \ln(v_{at_j}) + \beta_{sa_i, at_j} \quad (10)$$

Coordination Protocol (1/2)

- States the rules of interaction between ca and da_i
- proposed to satisfy the specific requirements of elasticity management negotiation
- Illocutions sent from da_i to ca are different from those sent from ca to da_i

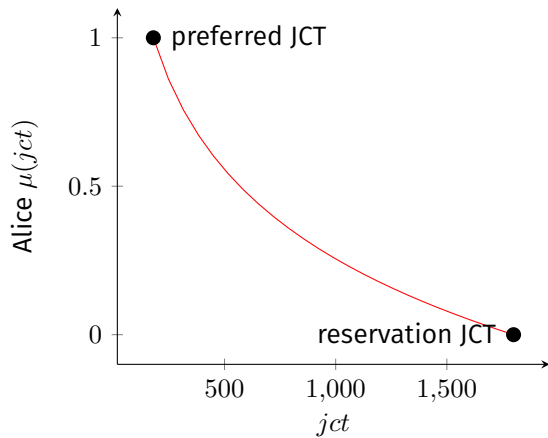
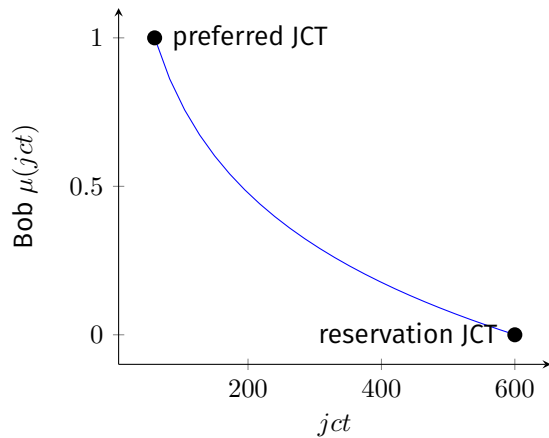
Illocutions ($da_i \rightarrow ca$)	Description
Inform ($event$)	da_i notifies the coordinator ca about an important $event$
RequestIntervention	da_i asks ca to intervene in its session i because the decision is beyond da_i capacities.

Coordination Protocol (2/2)

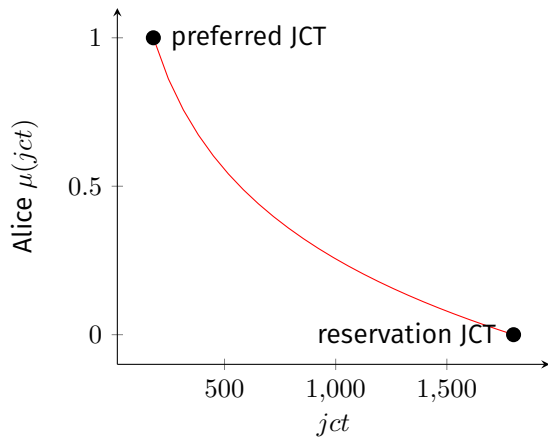
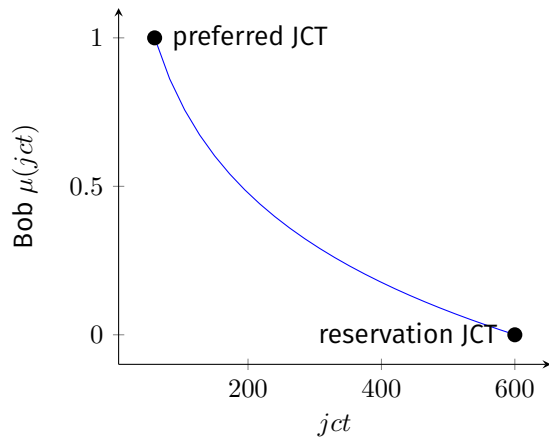
- The illocutions sent from the ca to da_i are *imperative*
- da_i cannot reject it

Illocutions ($ca \rightarrow da_i$)	Description
Spawn (Ω_{da_i})	ca spawns a delegate da_i to negotiate with sa_i
Modify (Ω_{da_i})	ca alters the negotiation strategy of da_i .
Suspend	ca orders the delegate da_i to suspend the negotiation process in the session i .
Kill	ca terminates the delegate da_i

Example

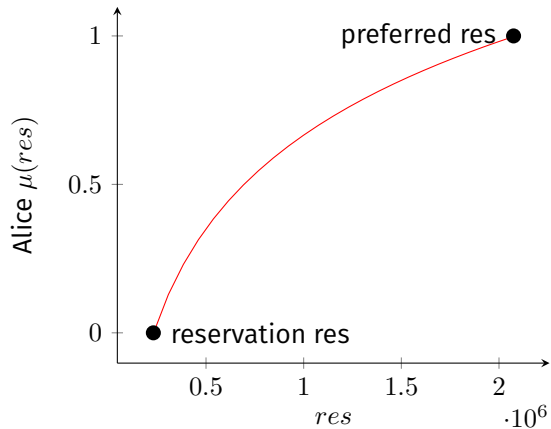
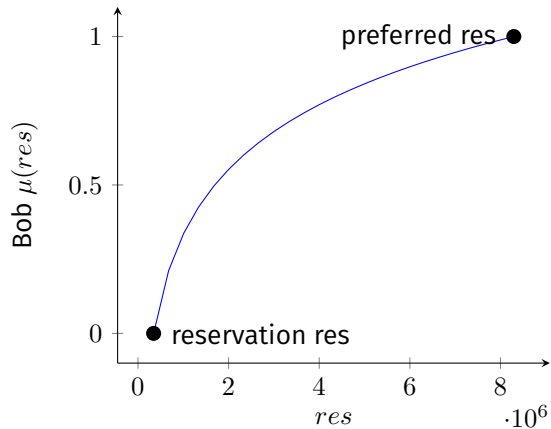


Example

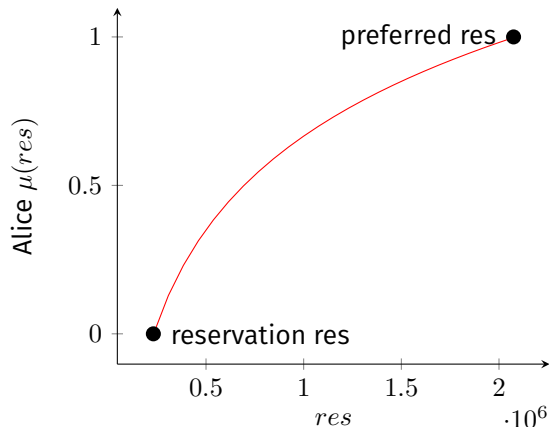
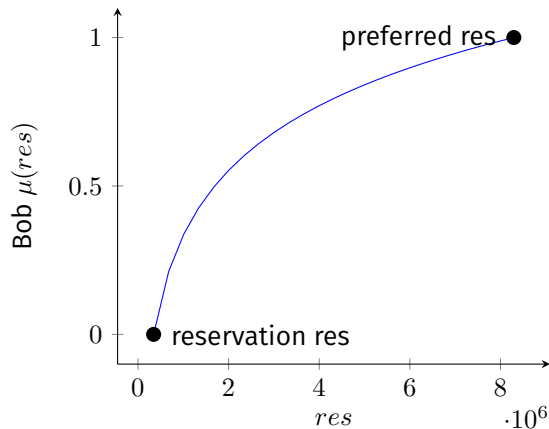


- $\mu_{sa_i, jct}$ is a decreasing logarithmic function
- Bob, gold SU, has lower reservation and preferred values
- Alice, bronze SU, has higher reservation and preferred values

Example (2)



Example (2)



- $\mu_{sa_i, res}$ is an increasing logarithmic function
- Bob, gold SU, has higher reservation and preferred values
- Alice, bronze SU, has lower reservation and preferred values
- M_{sa_i} , the overall utility function, is a weighted sum of $\mu_{sa_i, res}$ and $\mu_{sa_i, jct}$

Performance Model Example

Simple example of a performance model of a transcoding service

t/q	SD	MD	HD
10 min	4	5	8
20 min	2	4	6
30 min	1	2	3

- Tradeoffs
- Faster process requires more cloud resources \Rightarrow more costly for the ASP