

Calling the cloud: Enabling mobile phones as interfaces to cloud applications

Ioana Giurgiu, Oriana Riva, Dejan Juric, Ivan Krivulev, Gustavo Alonso

Systems Group, Computer Science Department, ETH Zurich



Motivation

Vision

- ▶ Mobile phones are set to become the universal interface to online services and cloud computing applications

What we have now

- ▶ Applications running on mobile phones are currently limited to two configurations
 - ▶ Run entirely on the phone
 - ▶ Access a centralized online service (web browser)



Motivation

Vision

- ▶ Mobile phones are set to become the universal interface to online services and cloud computing applications

What we have now

- ▶ Applications running on mobile phones are currently limited to two configurations
 - ▶ Run entirely on the phone
 - ▶ Access a centralized online service (web browser)



⇒ Is this a good model for phone-cloud applications?

Phone - cloud applications

Desired properties

- ▶ Single usage model
 - ▶ No preinstallation
 - ▶ Acquire on-the-fly
 - ▶ Discard after interaction
- ▶ Phone customization
 - ▶ Match phone constraints and user needs
- ▶ Application development
 - ▶ Write once, run everywhere

Scenario - Ticketing machine

- ▶ *Opportunity : Virtualize physical services*
- ▶ Goal
 - ▶ Turn the phone into an interface for the ticket machine
- ▶ Advantages
 - ▶ More interaction on the phone
 - ▶ More privacy
 - ▶ Combined functionalities
 - ▶ Ticket locally on the phone
 - ▶ Scales with multiple clients
- ▶ Challenges
 - ▶ Hardware diversity and heterogeneous user interfaces
 - ▶ Short-term, spontaneous interactions



More scenarios

Ma maison.ch3d - Sweet Home 3D

Fichier Éditeur Meubles Plan Vue 3D Aide

IKEA

Salle de bain
Salon

- Aquarium
- Bibliothèque
- Bibliothèque garnie
- Bureau
- Bureau PC

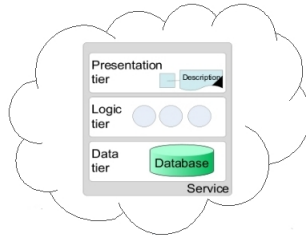
Nom	A	Largeur	Prof...	Hau...	Wei...
Canapé	147	87	87		☑
Chair	40	42	90		☑
Cuisin...	60	62	95		☑
Doubl...	132	34	210		☑
Evier	120	64	108		☑
Fenêtr...	132	34	117		☑
Lit 140	158	208	70		☑
Meubl...	101,3	64	95		☑
Meubl...	100	52	93		☑
Plante	58	50	82		☑
Porte	91,5	25,1	108,5		☑
Porte	91,5	25,1	108,5		☑

AlfredO

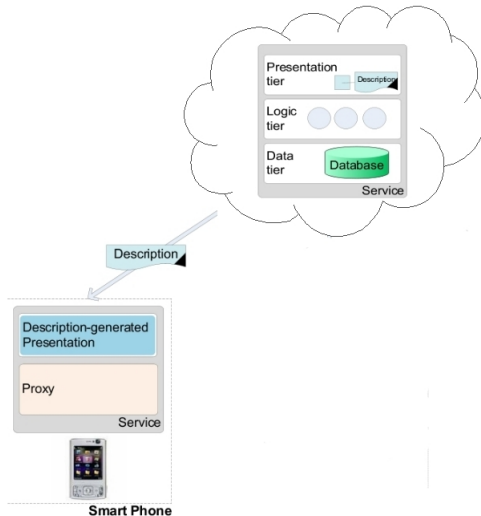
[Middleware'08]

- ▶ Supports partition and distribution of modularized applications in a client-server setup
 - ▶ OSGi-based
 - ▶ Focus on presentation and logic modules
- ▶ Flexible: a client can dynamically acquire different parts of an application
- ▶ Lightweight: memory footprint of less than 300kB
- ▶ Scales well with multiple concurrent interactions

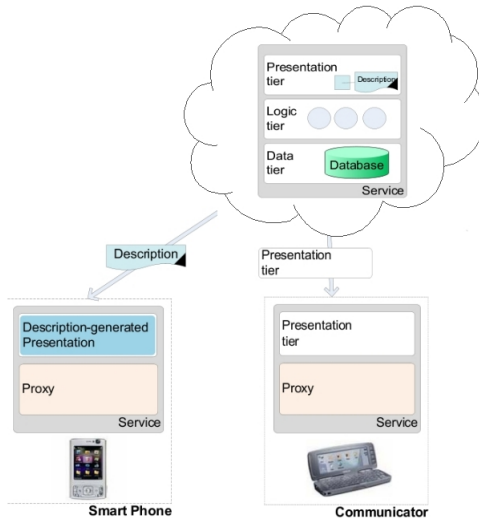
AlfredO



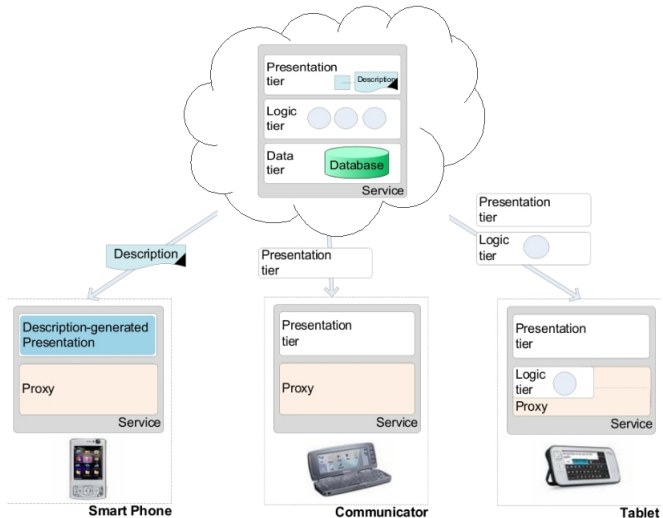
AlfredO



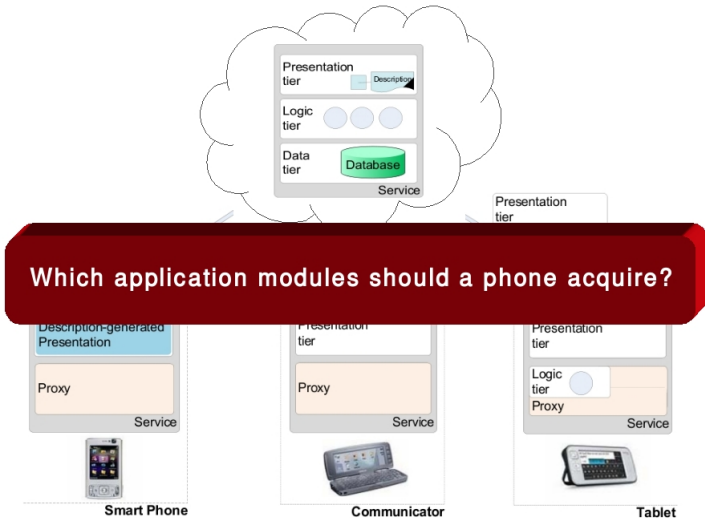
AlfredO



AlfredO



AlfredO



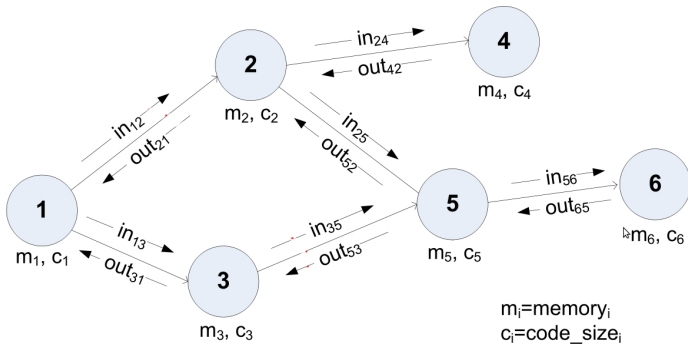
Optimization problem

- ▶ Premise: Modularized application
- ▶ Model of resource consumption and dependencies among application modules

Goal: Partition the application between phone and server based on different criteria

Application profiling

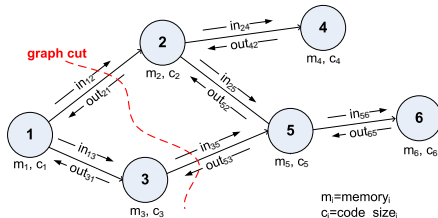
- ▶ Application is represented as *consumption graph*
- ▶ Model bandwidth and memory costs
- ▶ Lightweight modules



Application partitioning

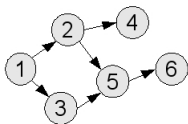
- ▶ Constraints
 - ▶ MAX(memory)
 - ▶ MAX(code_size)

- ▶ Objective
 - ▶ Minimize interaction time

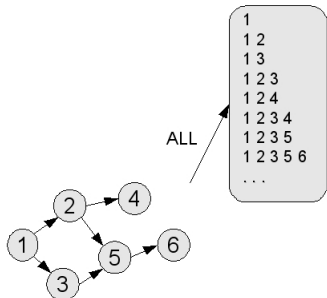


$$\min O_{C_c} = \min \left(\sum_{i=1}^{t < k} \sum_{j=1}^w \frac{(in_{ij} + out_{ji}) * f_{ij}}{\alpha} + \sum_{i=1}^k \frac{\text{code_size}_i}{\beta} + \sum_{i=1}^{w < s} \text{proxy_cost}_i \right)$$

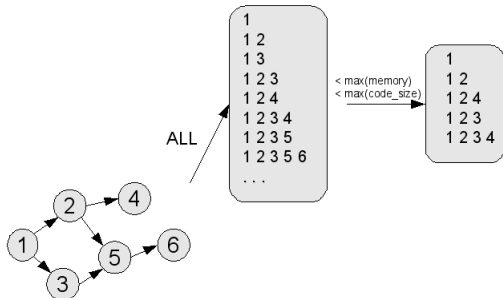
ALL and K-STEP algorithms



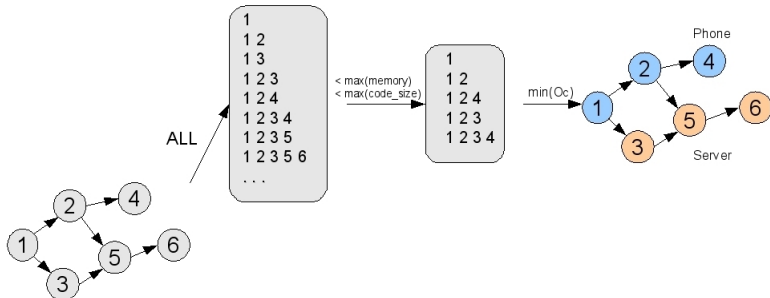
ALL and K-STEP algorithms



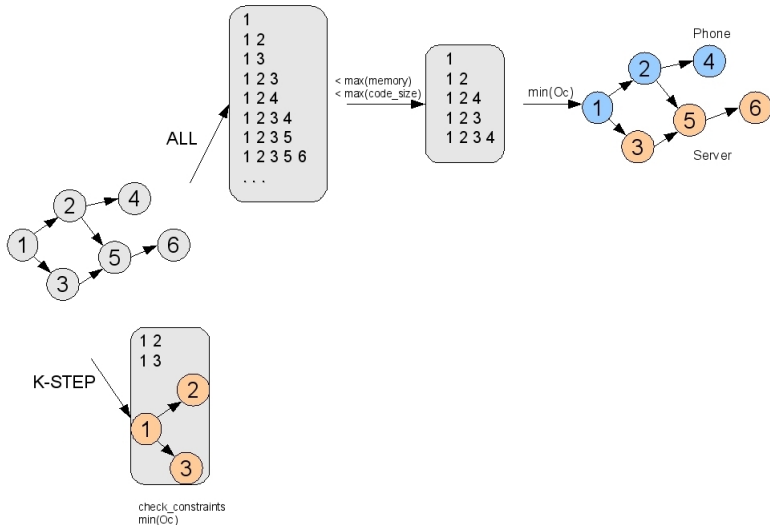
ALL and K-STEP algorithms



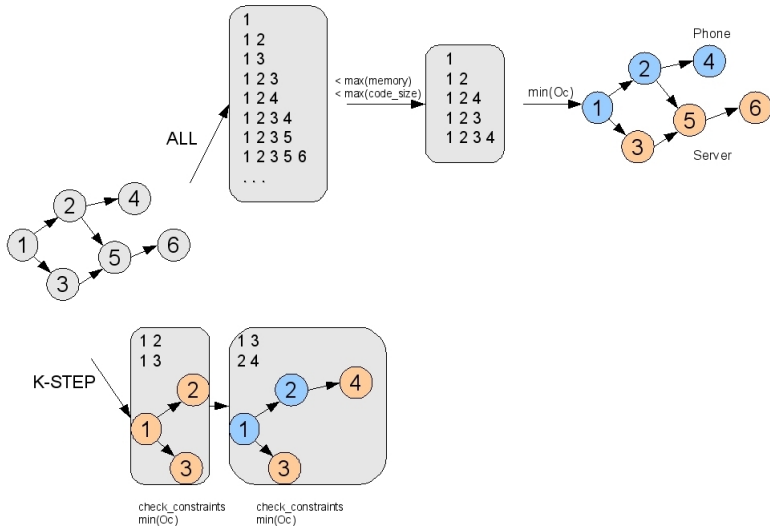
ALL and K-STEP algorithms



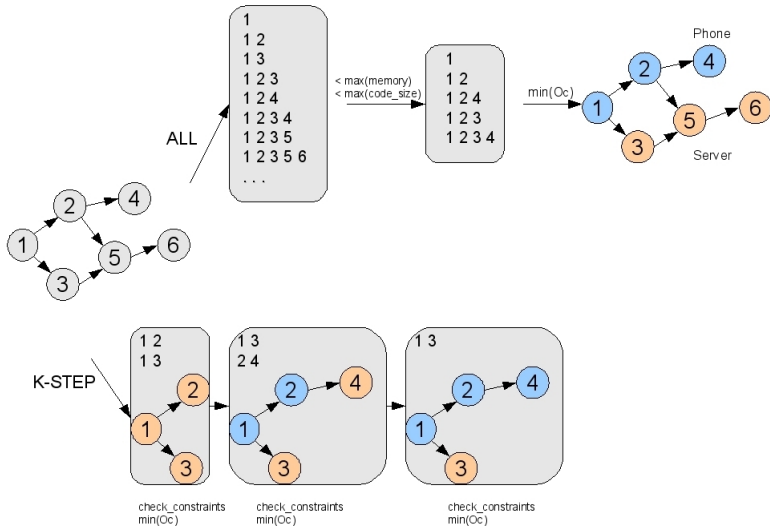
ALL and K-STEP algorithms



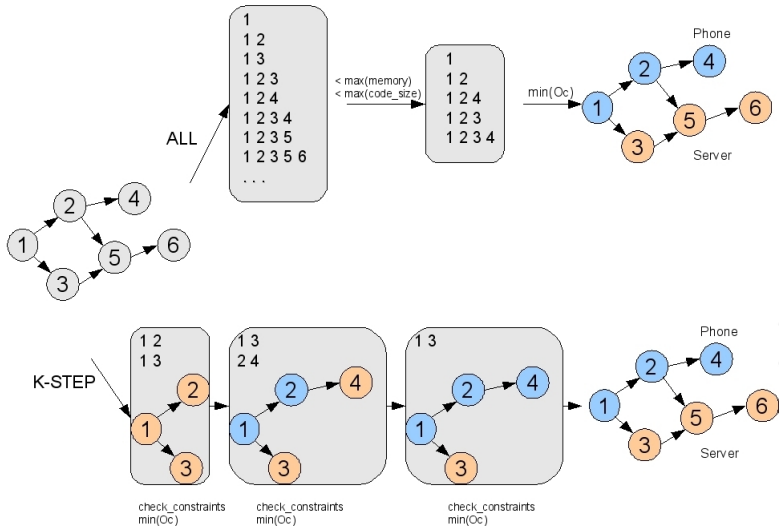
ALL and K-STEP algorithms



ALL and K-STEP algorithms

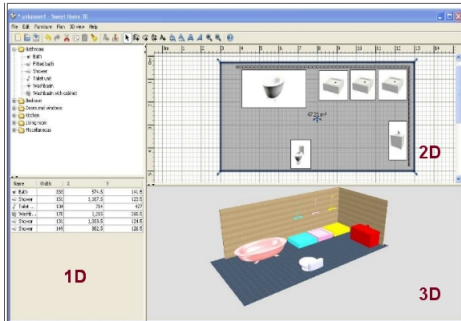


ALL and K-STEP algorithms



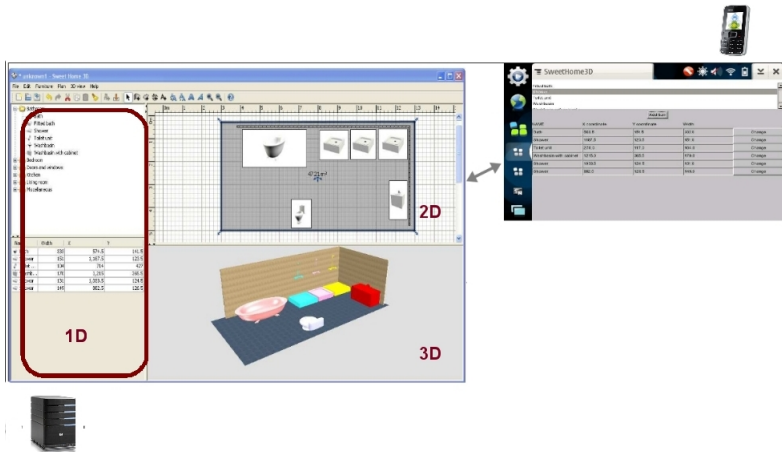
Applications

- ▶ Built from scratch (image processing)
- ▶ Existing ones (SweetHome3D)



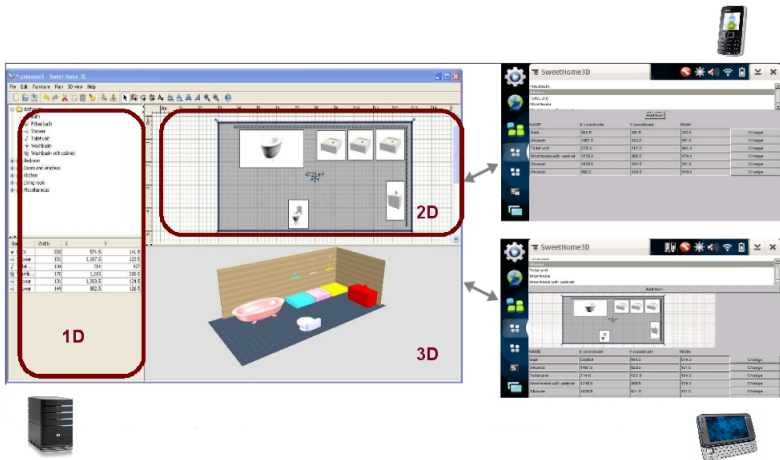
Applications

- ▶ Built from scratch (image processing)
- ▶ Existing ones (SweetHome3D)



Applications

- ▶ Built from scratch (image processing)
- ▶ Existing ones (SweetHome3D)



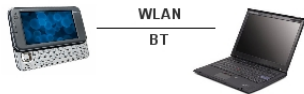
Evaluation

Optimization time

Startup time

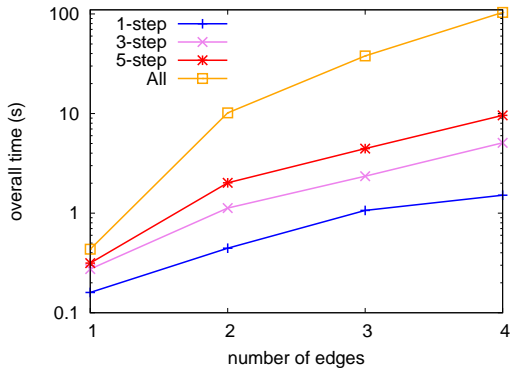
Interaction time

- ▶ Nokia N810 internet tablet
- ▶ ThinkPad T61p 2.5GHz

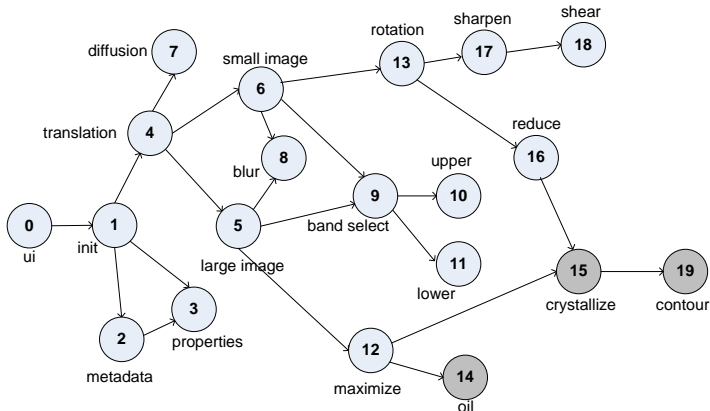


Evaluation - Optimization time

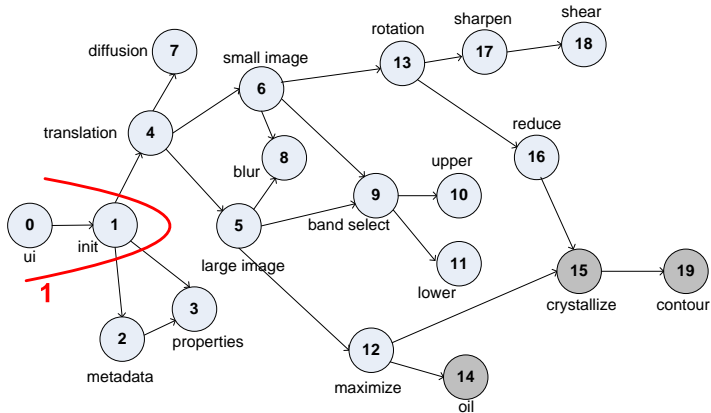
- ▶ ALL (global optimization)
 - ▶ Can be slow
- ▶ K-STEP (local optimization)
 - ▶ $\text{Max}(\text{err}) = 32\%$
 - ▶ Time increases with K



Evaluation - Prototype application

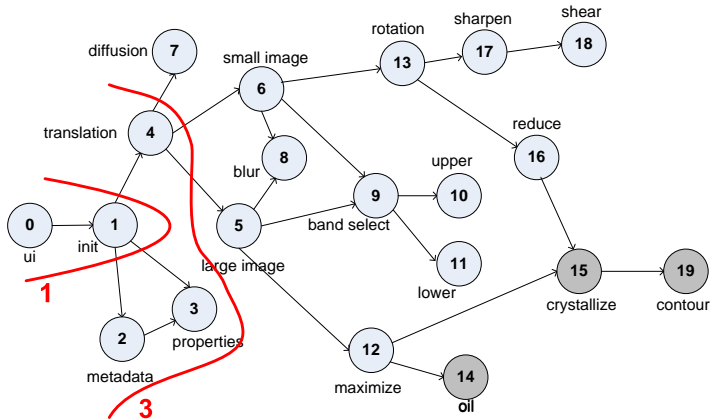


Evaluation - Startup time



► 1: 5.8 s

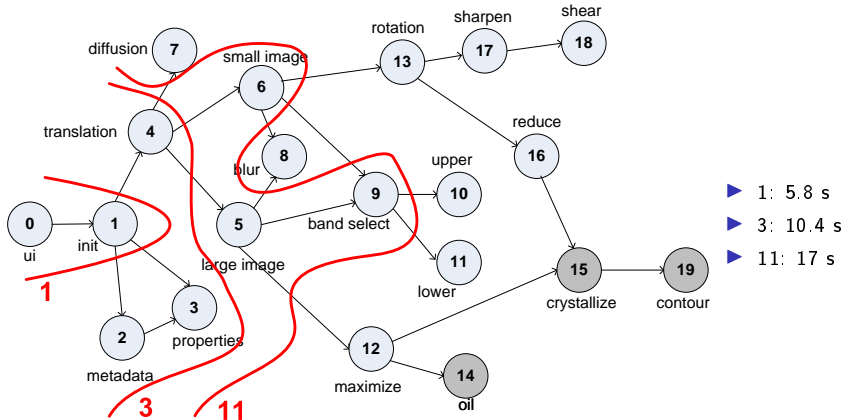
Evaluation - Startup time



▶ 1: 5.8 s

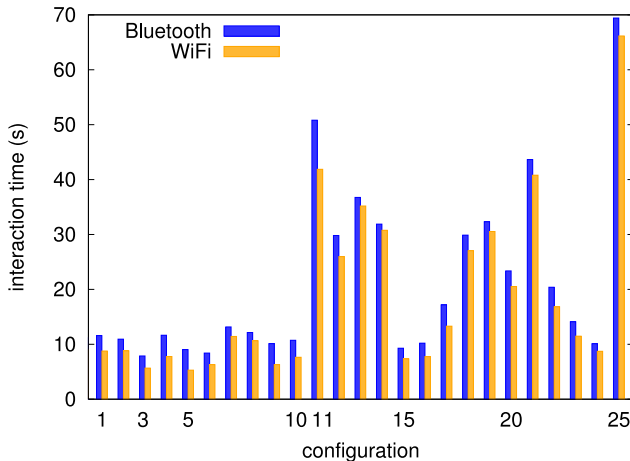
▶ 3: 10.4 s

Evaluation - Startup time



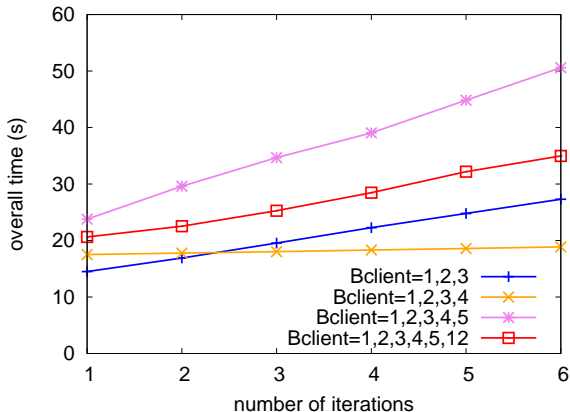
Evaluation - Choosing the right configuration matters

- ▶ Space for improvement (seconds / tens of seconds)



Evaluation - Continuous interaction

- ▶ When is it convenient to acquire a service invoked multiple times?



Conclusions

- ▶ SaaS distribution model for physical services
- ▶ Phones become universal interfaces to cloud applications
- ▶ Optimized and dynamic distribution of application modules
- ▶ Phones achieve better performance with a controlled overhead

Current focus

- ▶ Complete platform that facilitates the development of modularized applications in phone - cloud setups