Auction-Based Resource Allocation in Digital Ecosystems

Moreno Marzolla, Stefano Ferretti, Gabriele D'Angelo

Dipartimento di Informatica—Scienza e Ingegneria (DISI)
Università di Bologna, Italy

moreno.marzolla@unibo.it  sferrett@cs.unibo.it  g.dangelo@unibo.it
Talk Outline

• Motivation
  – Digital Organism (DO)
  – Digital Ecosystem (DE)
• Resource sharing in a DE using ascending-price auctions
• Performance evaluation
• Conclusions and future works
Scenario

- A medical doctor receives an urgent call during a meeting
- She needs to check some medical data to diagnose a particular illness
- Her tablet is not powerful enough for that
- She “rents” CPU power from one of his colleagues high-end laptops to analyze the data

http://www.flickr.com/photos/dellphotos/10347671333/
Another Scenario

- A user wants to upload some pictures to Flickr, but lacks network connectivity
- He exploits the 3G mobile connection of a neighboring friend through an ad-hoc short range connection (e.g., bluetooth)

http://www.flickr.com/photos/mjm/184754322/
A “Digital Organism” (DO)

- A mobile user is considered a “Digital Organism” (DO) composed of different devices.
- Devices provide computation, communication or storage resources.

Digital Ecosystem

- DOs can share resources opportunistically
- Smart P2P schemes can be used among DOs to share data and resources
- A community of interacting DOs can be considered as a “Digital Ecosystem” (DE)
Digital Ecosystem
Digital Ecosystem

• Many problems to be addressed
  – How is the DE organized? What is its topology? How can resources be accessed by other DOs? What access control schemes have to be used? How can peers be trusted? ...

• In this presentation we address one of those problems
  – How can resources be allocated among requesting DOs?
Problem Formulation

• Given:
  – $N$ users
  – $R$ resource types (e.g., “CPU”, “Network”, “Storage”)
  – User $i$ requests $\text{Req}_{r,i}$ units of resource $r$
  – User $j$ offers $\text{Off}_{r,j}$ units of resource $r$
  – $M_{i,j} = 1$ iff user $i$ can communicate with user $j$

• Maximize the number of matching request-offer pairs among adjacent users
  – Each user must get all the resources it requires, or none at all
Problem formulation

- The problem can be formulated as a Mixed Integer Linear Programming (IP) problem
  - See backup slides
- **The Good**
  - The IP problem can be used to compute the optimal matching
- **The Bad**
  - IP is in general NP-complete
- **The Ugly**
  - Global knowledge of the whole system is required
  - Fairness is not taken into account


MOBILWARE 2013, Bologna, Italy
Auction-Based Resource Allocation

• Fully distributed resource allocation mechanism
• Each user is given some amount of tokens that can be used to “buy” resources
  – Tokens provide an economic incentive to share resources and avoid free-riding
  – Some external mechanism to generate and transfer tokens securely is required; we don't address this problem here
• Each user engages in an auction with her neighbors to acquire/sell resources usage
• The same user can play the role of seller and buyer at the same time
Auction-Based Resource Allocation

• Sellers advertise the unitary prices of the resources they sell
  – A seller increases the price of all resources for which he is unable to fulfill all requests (Ascending-Bid Auction)

• A buyer can acquire different resource types from different sellers
  – However, each buyer must acquire all instances of any resource type from the same seller
  – Resources can not be partitioned across different sellers
User 1 can interact with User 3 and User 4

User 1

Reserve Price: 15€
Request: (3, 6)

Offer: (5, 3)
Sell Prices: (1€, 1€)

User 2 can interact with User 4 only

User 2

Reserve Price: 10€
Request: (2, 6)

Offer: (2, 10)
Sell Prices: (1€, 1€)
Example

User 3 can only provide enough resource 1, while User 4 can only provide enough resource 2. Therefore, I will bid User 3 for res 1 and User 4 for res 2.

I can interact with User 4 only. Luckily, User 4 can potentially fulfill my whole request. I will bid User 4.

Reserve Price: 15€
Request: (3, 6)

Bid (3, -)
Bid (-, 6)

Offer: (5, 3)
Sell Prices: (1€, 1€)

Offer: (2, 10)
Sell Prices: (1€, 1€)

Reserve Price: 10€
Request: (2, 6)

Bid (2, 6)

User 1

User 2

User 3

User 4
Example

User 1

Reserve Price: 15€
Request: (3, 6)

Bid (3, -)

Offer: (5, 3)
Sell Prices: (1€, 1€)

User 3

I have enough resources to fulfill all requests. I confirm the selling price.

User 2

Reserve Price: 10€
Request: (2, 6)

Bid (-, 6)

Bid (2, 6)

Offer: (2, 10)
Sell Prices: (1€, 1€)

User 4

I don't have enough resource 2 to fulfill all requests. I increase the price of resource 2, let's see what happens.
Example

User 1

Reserve Price: 15€
Request: (3, 6)

Price: (1€, -)
Offer: (5, 3)
Sell Prices: (1€, 1€)

User 2

Reserve Price: 10€
Request: (2, 6)

Price: (-, 2€)
Offer: (2, 10)
Sell Prices: (1€, 2€)

User 3

User 4
Example

If I accept the proposed price, I will pay 15€, which is within my reserve price. I bid again.

User 1

Reserve Price: 15€
Request: (3, 6)
Price: (1€, -)
Offer: (5, 3)
Sell Prices: (1€, 1€)

User 2

Reserve Price: 10€
Request: (2, 6)
Price: (-, 2€)
Offer: (2, 10)
Sell Prices: (1€, 2€)

If I accept the proposed price, I will pay 14€, which is above my reserve price. I give up.

User 3

User 4
Example

User 1

Reserve Price: 15€
Request: (3, 6)

Bid (3, -)

Offer: (5, 3)
Sell Prices: (1€, 1€)

User 2

Reserve Price: 10€
Request: (2, 6)

Bid (-, 6)

Offer: (2, 10)
Sell Prices: (1€, 2€)

User 3

User 4

I have enough resources to fulfill all requests. I confirm the selling price.

I have enough resources to fulfill all requests. I confirm the selling price.
Example

User 1

Reserve Price: 15€
Request: (3, 6)
Price: (1€, -)
Offer: (5, 3)
Sell Prices: (1€, 1€)

User 2

Reserve Price: 10€
Request: (2, 6)
Price: (-, 2€)
Offer: (2, 10)
Sell Prices: (1€, 2€)

User 3

User 4

Good, bid accepted
Pricing mechanism

• Each seller defines:
  – his initial selling price for each resource offered
  – the price increment for each round

• Each buyer defines his reserve price

• Caveats
  – If the initial selling price or the price increment is too high, then the seller may be unable to sell even if there are potential buyers
  – If the price increment is too small, auctions may require a large number of iterations to conclude
Evaluation

- \( N = \{10, 20, 50\} \) users
- Total initial budget for each user: 100 tokens
- \( R = \{3, 5, 7\} \) resource types
- Mean connection density 30%
- \( T = 10 \) auctions; for each auction:
  - 20% of users are randomly selected as pure buyers, the other users are pure sellers
  - Random vectors of requests/offers
  - Initial unitary price for sellers: 1 token
  - Randomly generated reserve price for buyers
- Each experiment is repeated 20 times
- The IP model is used as the baseline (ignoring prices and budgets)
Mean number of matches at the end of all auctions
Behavior on crowded markets

- $N = 50$ users, $R = 7$ resource types
- Connection densities $\rho = \{0.2, 0.4, 0.6, 0.8\}$
Behavior on crowded markets

- Each buyer always bids the lowest price
- When $\rho$ becomes large, the same “most convenient” seller may be contended by many potential buyers
Behavior on crowded markets

Average unitary price of resources

Average number of rounds per auction

Connection density ($\rho$)
Conclusions

• We addressed the problem of resource sharing between “Digital Organisms” operating in a heterogeneous “Digital Ecosystem”
• We proposed a fully decentralized algorithm for resource sharing based on ascending-bid auctions
• The algorithm requires local interactions only, and can produce good solutions in realistic scenarios
• The auction algorithm requires an underlying system of economic incentives to ensure fairness and avoid free riding
Open Issues

- Token generation and distribution
- Privacy, accountability and trust
- The proposed auction scheme does not take into consideration QoS attributes
  - A seller may provide different instances of the same resource type, at different prices (e.g., 3G vs WiFi connectivity)
  - A buyer may have different reserve prices for different QoS values
- The implementation of a prototype running on mobile phones is in the planning phase
Thanks for your attention!

Questions?

Moreno Marzolla
Dipartimento di Informatica—Scienza e Ingegneria (DISI)
Università di Bologna
moreno.marzolla@unibo.it
http://www.moreno.marzolla.name/
Backup Slides
IP problem formulation

• Given:
  - \( N := \{1, \ldots, N\} \) set of users
  - \( R := \{1, \ldots, R\} \) set of resources
  - \( \text{Req } r_i := \text{Amount of resource } r \text{ requested by } i \)
  - \( \text{Off } r_j := \text{Amount of resource } r \text{ offered by } j \)
  - \( M_{ij} := 1 \text{ iff user } i \text{ can interact with user } j \)

• Define
  - \( X_{irj} := 1 \text{ iff user } i \text{ gets resource } r \text{ from user } j \)
IP problem formulation

• Maximize
  \[
  \sum_{i \in N} \sum_{r \in R} \sum_{j \in N} X_{irj}
  \]

• Subject to
  \[
  \sum_{i \in N} \text{Req}_{ri} X_{irj} \leq \text{Off}_{rj} \quad r \in R, j \in N
  \]
  \[
  \sum_{j \in N} X_{irj} \leq 1 \quad i \in N, r \in R
  \]
  \[
  \sum_{j \in N} X_{irj} \leq \sum_{j \in N} X_{ilj} \quad i \in N, r \in R
  \]
  \[
  X_{irj} \leq M_{ij} \quad i \in N, r \in R, j \in N
  \]
IP problem formulation

- Maximize

\[
\sum_{i \in N} \sum_{r \in R} \sum_{j \in N} X_{irj}
\]

- Subject to

\[
\sum_{i \in N} \text{Req}_{ri} X_{irj} \leq \text{Off}_{rj}, \quad \forall r \in R, \quad \forall i \in N
\]

\[
\sum_{j \in N} X_{irj} \leq 1, \quad \forall i \in N, \quad \forall r \in R
\]

\[
\sum_{j \in N} X_{irj} \leq \sum_{j \in N} X_{ilj}, \quad \forall i \in N, \quad \forall r \in R
\]

\[
X_{irj} \leq M_{ij}, \quad \forall i \in N, \quad \forall r \in R, \quad \forall j \in N
\]

- The total amount of resource \( r \) requested to node \( j \) must not exceed the quantity it offers.
- Each node \( i \) must obtain resource \( r \) by at most a single provider.
- Each node \( i \) must obtain all the resources it requests, or none at all.
- Node \( i \) can get anything from node \( j \) only if \( i \) and \( j \) are neighbors.