IEEE International Conference on Cloud and Service Computing (CSC 2011)

Social-optimized Win-win Resource Allocation for Self-organizing Cloud

Sheng Di, Cho-Li Wang, **Luwei Cheng**, Lin Chen
The University of Hong Kong

Dec. 14, 2011
Hong Kong, China

Outline

- Background
- Research Motivation
  - The economy of P2P Cloud
- System Model
- Research Problems
  - Resource Discovery: Multi-dimension Range Query over DHT
  - Resource Allocation: Dual-Vickrey Auction (DVA) algorithm
- Performance Evaluation
- Related Work
- Conclusion and Future Work

Background

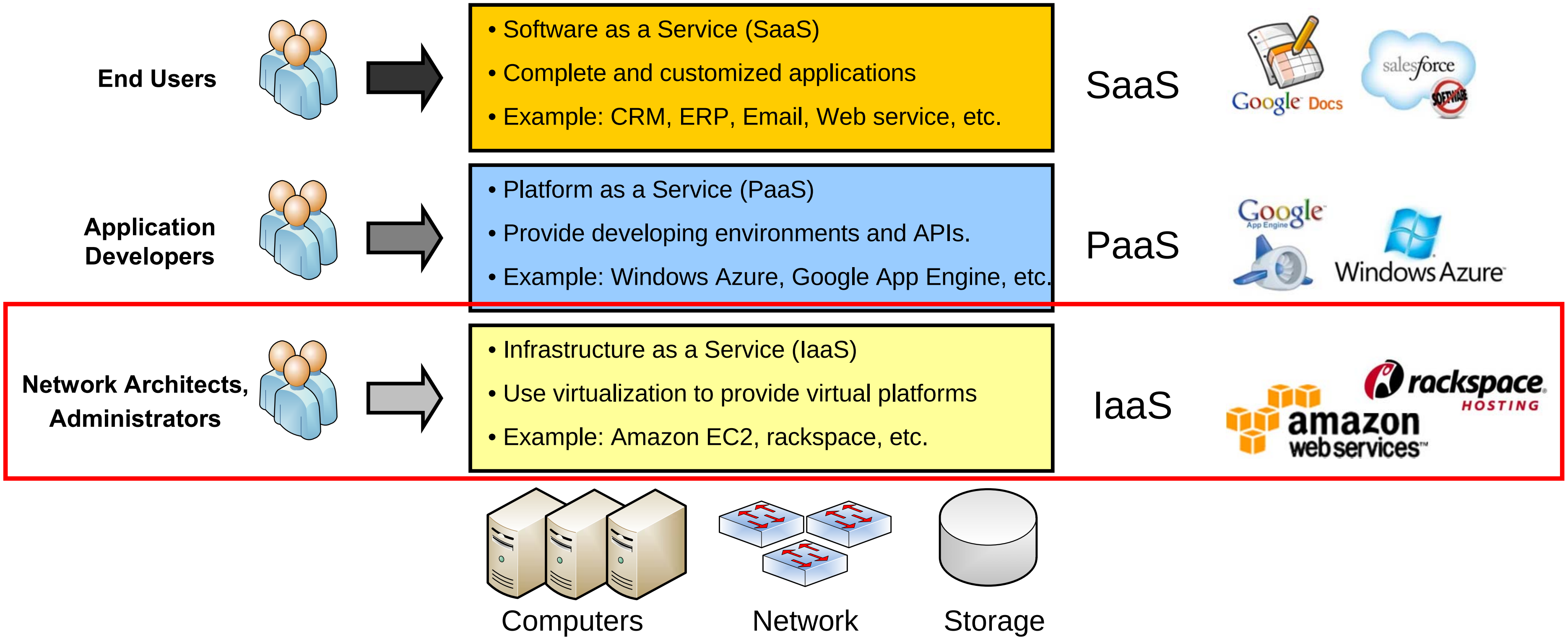
- Definition of Cloud Computing from National Institute of Standards and Technology (NIST)
  - **Cloud computing** is defined as "a model for enabling **ubiquitous, convenient, on-demand network access** to a **shared** pool of configurable computing resources (for example **networks, memories, storage, CPU, services, etc.**) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

Elasticity (virtual machines)

Multi-dimensional

Background

- Service Model of Cloud Computing



End Users

- Software as a Service (SaaS)
- Complete and customized applications
- Example: CRM, ERP, Email, Web service, etc.

SaaS

Application Developers

- Platform as a Service (PaaS)
- Provide developing environments and APIs.
- Example: Windows Azure, Google App Engine, etc.

PaaS

Network Architects, Administrators

- Infrastructure as a Service (IaaS)
- Use virtualization to provide virtual platforms
- Example: Amazon EC2, rackspace, etc.

IaaS

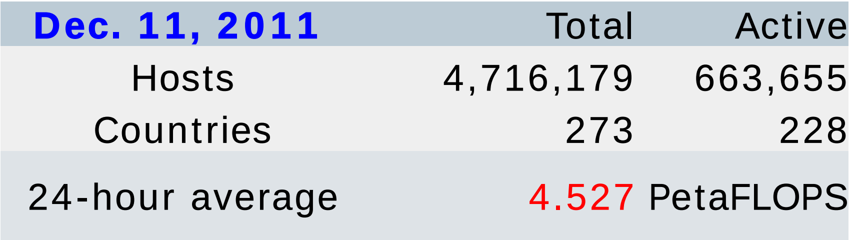
Computers

Network

Storage

Why P2P Cloud?

- Internet resources are MASSIVE



Dec. 11, 2011	Total	Active
Hosts	4,716,179	663,655
Countries	273	228
24-hour average	4.527 PetaFLOPS	

K computer
Rmax: 8.162 PFLOPS

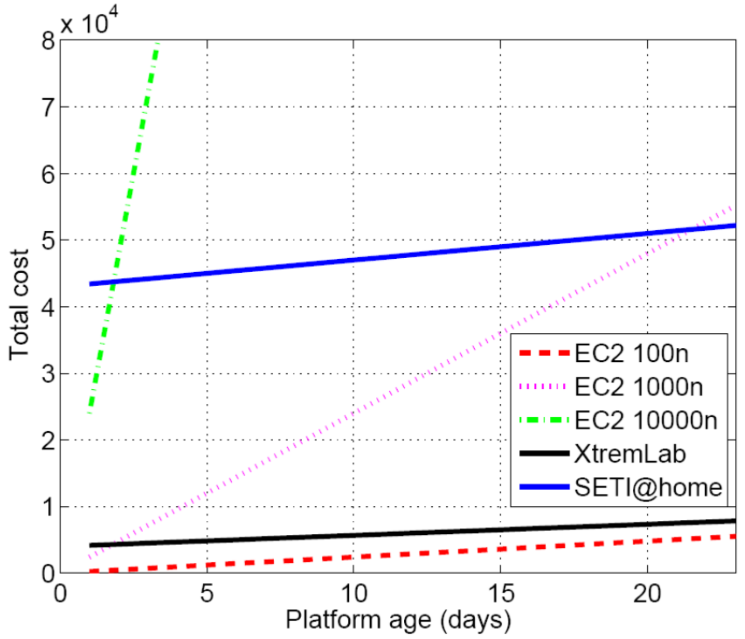
Storing 150 million files distributed among Internet hosts, by April 2010

Why P2P cloud?

- Volunteer Cloud is much CHEAPER

- SETI@Home:
  - Cost-benefit desktop grids
  - More than 350000 nodes
- XtremLab:
  - a BOINC-based project that actively measures host CPU and network availability on volunteer desktops
- Cloud@Home:
  - Bridge the gap between volunteer and cloud
  - Started in INRIA (France) from 2009
- Nebulas:
  - Use distributed voluntary resources to build cloud
  - University of Minnesota (HotCloud09)

Total cost of Cloud versus VC over time



Source: “Cost-Benefit Analysis of Cloud Computing versus Desktop Grids” (HCW09)

Research Motivation

- Other benefits of P2P Cloud:
  - Decentralized organization of resources over internet
    - Avoid single-point-of-failure risk, DDoS attacks on WikiLeaks (29/11/2010).
  - Self-organizing
    - Less management effort
  - Locality-aware
    - Better service quality
  - Cooperative community
    - Better use of idle resources
  - … …

Big cloud goes distributed

- Content Delivery Network (CDN): The “BIG CLOUD” use millions of distributed servers to provide service
  - Akamai (2011): 95,000 servers, 71 countries.
  - Google (2011): 1 million servers, 40 global datacenters.
  - Amazon (2011): 50,000 servers, 53 edge locations.
  - Yahoo, Microsoft, ….
  - PlanetLab: world-wide Internet test bed

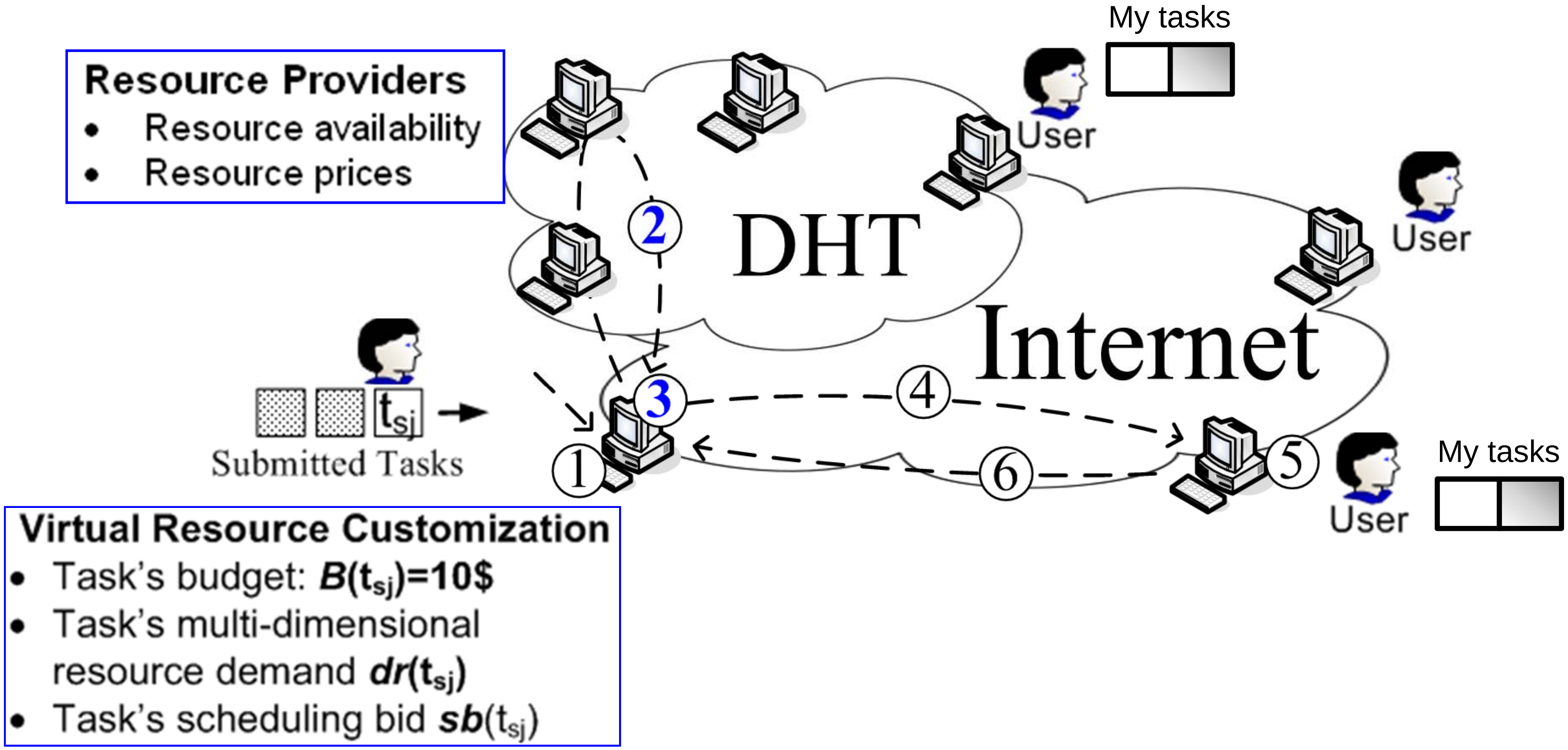
- Benefits of distributed cloud:
  - Fast access speed by local users (cache servers)
  - Better fault tolerance across datacenters

Research Challenge

- The incentive is the key!
  - Why should people contribute resources to P2P cloud?
  - Why should people use P2P cloud?

- From an economic perspective, we propose a model to encourage the involvement of P2P Cloud
  - A win-win situation
    - Let both resource providers & consumers be satisfied with final gains, so they are willing to participate in the cloud
  - Social welfare
    - From a global viewpoint, the total resource utility should be optimized to guarantee the overall performance, with fairness kept.

System Model



Resource Providers

- Resource availability
- Resource prices

Virtual Resource Customization

- Task's budget: $\boldsymbol{B}(\mathbf{t_{sj}})\mathbf{=10\$}$
- Task's multi-dimensional resource demand $\boldsymbol{dr}(\mathbf{t_{sj}})$
- Task's scheduling bid $\boldsymbol{sb}(t_{sj})$

(1) Task Submission; (2) Range-query for Qualified Resources (3) Task assignment based on resources found; (4) Task migration; (5) Execute the task; (6) Task-finish notification with result returned.

Research Problems

- In self-organizing architecture, every host autonomously manages its tasks. Two problems must be effectively addressed:
  - Resource Discovery Protocol
    - How to design an effective discovery protocol to find qualified resources in P2P environment?
    - CAN? Chord? Pastry? Gnutella?
  - Resource Allocation Method
    - There may be many qualified resource providers. How to schedule and execute the tasks under users' demand and resource suppliers' expectation on their payoffs?
    - User A demands resources under limited budget; User B wants to sell resources in expected prices …

Resource Discovery Protocol

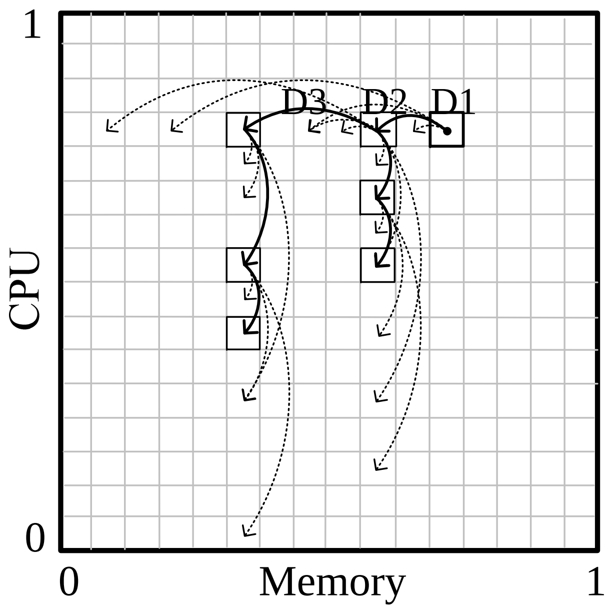
- In P2P cloud system, it desires multi-dimensional range-search to discover the resources:
  - Not single-dimension single-value search
    - Dimension 1: CPU capability is within [c1, c2];
    - Dimension 2: Memory is within [m1, m2];
    - Dimension 3: storage;
    - Dimension 4: bandwidth, latency, etc…
- The multi-dimensional range search problem is challenging:
  - Contentions along multiple dimensions could happen in the presence of the uncoordinated queries.
  - Low resource matching rate may happen while restricting query delay and network traffic.

Resource Discovery Protocol

- PID-CAN (Proactive Index Diffusion over CAN):
  - Proactively diffuse resource indexes over the nodes
  - Randomly route query messages among them

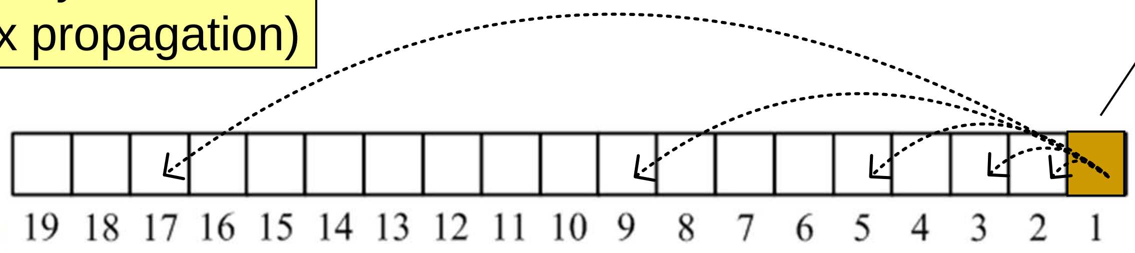
⇢ index nodes on track

⟶ randomly selected index nodes



Propagating my identifiers (a.k.a. index propagation)

I know some idle nodes



Resource Discovery Protocol

- PID-CAN features:
  - Stable and optimized searching performance
  - Low query delay
  - Low traffic overhead

- More details, please refer to the following paper:
  - "*Probabilistic Best-fit Multi-dimensional Range Query in Self-organizing Cloud*", in the 40th International Conference on Parallel Processing (ICPP 2011). Taipei, Taiwan, September 13-16, 2011.

Resource Allocation Algorithm

- Consideration:
  - There may be multiple qualified resource providers after resource discovery. How to schedule and execute the tasks under users' demand and resource suppliers' expectation on their payoffs?

- Goals:
  - A win-win situation
    - Let both resource providers & consumers be satisfied with final gains, so they have the incentive to participate in the cloud
  - Social welfare
    - From a global viewpoint, the total resource utility should be optimized to guarantee the overall performance (such as throughput), with fairness kept.

Problem Formulation

- Assuming there are n peer nodes: P_1, P_2, … , P_n. Each node can serve as both resource consumer and resource provider.

- Each task t_{ij} submitted to queue q_i of node P_i has: (1) resource demand $dr(t_{ij})$. (2) budget $B(t_{sj})$. (3) scheduling bid $sb(t_{ij})$. (4) resource price bid $sp(t_{ij})$

$$dr(t_{ij})=(dr_1(t_{ij}), dr_2(t_{ij}), \cdots, dr_R(t_{ij}))^T$$

- Let $c_k(P_i)$ denote the capacity of the *kth* resources attribute of node P_i, and $r(t_{ij})$ denote the actual amount of resources allocated to task t_{ij} .

$$r(t_{ij}) = (r_1(t_{ij}), r_2(t_{ij}), \cdots, r_R(t_{ij}))^T$$

- Resource availability states $a(P_i)$ will be propagated using PID-CAN protocol, for other nodes to discover the available resources.

$$a(p_d) = (a_1(p_d), a_2(p_d), \cdots, a_R(p_d))^T \quad a_k(p_d)=c_k(p_d)-\sum_{\forall i,j} dr_k(t_{ij}^{p_d})$$

Problem Formulation

- Assumption: some tasks’ characteristics can be predicted based on historical execution records, or analysis based on intrinsic programming structures [15].
  - With more resources, the task can be finished faster.
  - So as to save time, as long as the current budget allows:
    - Competition 1: The tasks will use high bid to compete for high scheduling priority (scheduling utility).
      - Some users may need faster response speed.
    - Competition 2: The tasks will request as more resources as possible to accelerate the execution (execution utility).
      - Use low price to buy more resources.

- Task’s utility = scheduling utility + execution utility

Problem Formulation

- Task utility: $tu(t_{ij})$
- Scheduling utility: $su(t_{ij})$
- Execution utility: $eu(t_{ij})$

$$\boxed{tu(t_{ij}) = \lambda_{ij} \cdot su(t_{ij}) + (1 - \lambda_{ij}) \cdot eu(t_{ij})}$$ (normalized)

λ_{ij} a coefficient customized based on user's expectation

- Objective function: Average Task Utility (Social Welfare)

$$\boxed{ATU = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m_i} tu(t_{ij})}{\sum_{i=1}^{n} m_i}}$$

- Constraints:

$$\sum_{i,j} dr_k(t_{ij}^{p_d}) \leq c_k(p_d), \quad k = 1, 2, \cdots, R \qquad (1)$$

$$dr_k(t_{ij}) \leq r_k(t_{ij}^{p_d}), \quad k = 1, 2, \cdots, R \qquad (2)$$

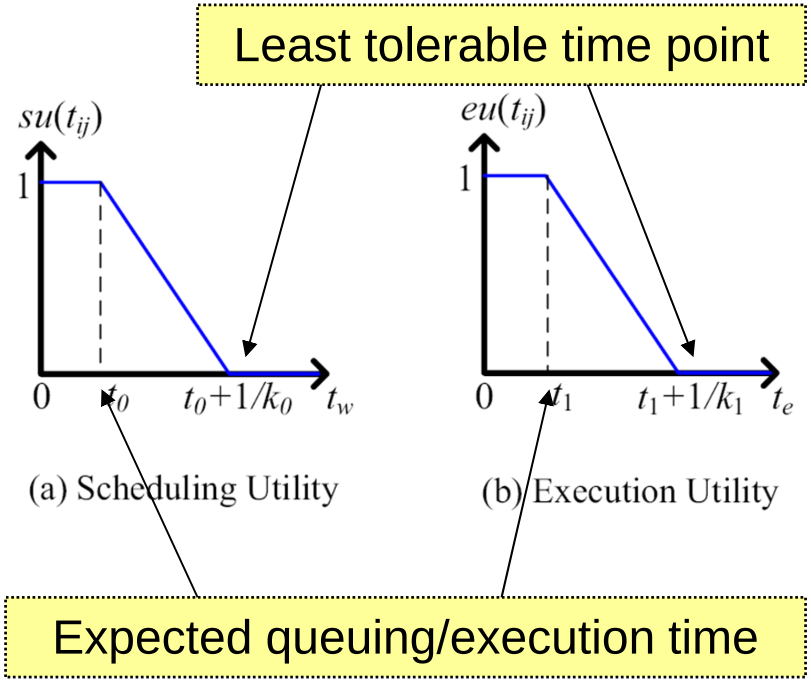
Problem Formulation

- Without loss of generality, we define *su*(t_{ij}): and *eu*(t_{ij}) to be two piecewise linear functions:

$$su(t_{ij}) = \begin{cases} 1 & t_w \leq t_0 \\ 1 - k_0(t_w - t_0) & t_0 < t_w \leq t_0 + 1/k_0 \\ 0 & t_w > t_0 + 1/k_0 \end{cases}$$

$$eu(t_{ij}) = \begin{cases} 1 & t_e \leq t_0 \\ 1 - k_1(t_e - t_1) & t_1 < t_e \leq t_1 + 1/k_1 \\ 0 & t_e > t_1 + 1/k_1 \end{cases}$$

Problem Formulation



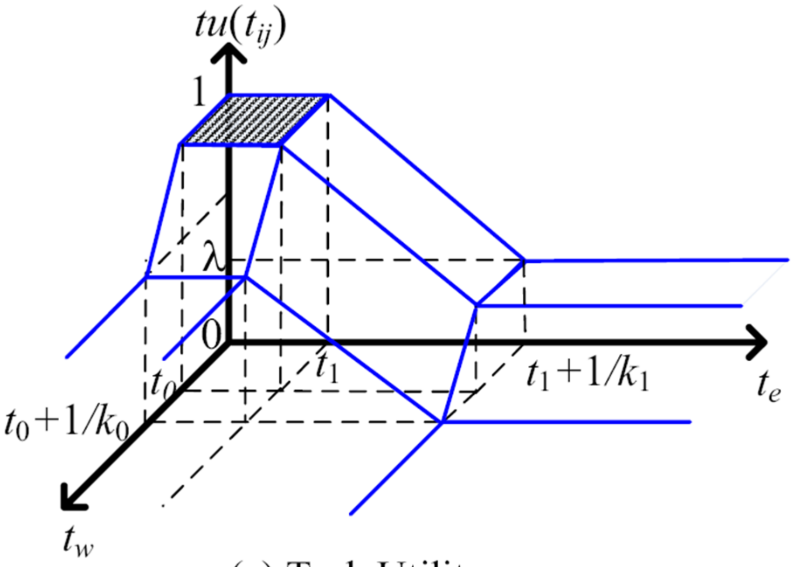
Least tolerable time point

(a) Scheduling Utility

(b) Execution Utility

Expected queuing/execution time

$$tu(t_{ij}) = \lambda_{ij} \cdot su(t_{ij}) + (1 - \lambda_{ij}) \cdot eu(t_{ij})$$



(c) Task Utility

Theoretical basis of solutions: Auction

- Open Auction (English auction and Dutch auction):
  - Time-consuming.
  - Potentially induce the resource price much higher than the real value, sacrificing auctioneers' benefits.

- Sealed Auction (First-price and Second-price):
  - It is fast.
  - How to guarantee the incentive compatibility
    - Each participant should always be honest on their true demands
  - First-price Auction:
    - Participates incline to bid the prices that are lower than the true value of the resource they regard, at the cost of resource owner's profit.
  - Second-price Auction:
    - Proved to achieve incentive compatibility. The participants who lie against their true demands will get inferior gains.

Theoretical basis: Vickrey Auction

- Vickrey Auction:
  - sealed-bid auction
    - bidders submit written bids without knowing the bid of the others
  - the highest bidder wins; the price paid is the second-highest bid.

- Why we choose “Vickrey Auction” for resource allocation?
  - In P2P environment, the peers are unknown to each other but they are competing with each other.
  - **The incentive to be honest**: each auctioneer is willing to reveal its true evaluations under Vickery Auction (strategy-proved)

- Vickrey Auction-based design is very suitable for P2P systems!

Resource Allocation: Dual-Vickrey Auction (DVA)

- Dual-Vickrey Auction (DVA):

```
Sort $q_s$'s tasks in non-increasing order of $sb(t_{sj})$;
for (each task $t_{sj}$ in $q_s$)
{
    $\max_{sb(txy) \le sb(tsj)} (sb(t_{xy})) \rightarrow sp(t_{sj})$ ;
    Perform PID-CAN to construct $QSET(t_{sj})$ for $t_{sj}$ ;
    Sort all items in $QSET(t_{sj})$ in non-decreasing
    order of $t_{sj}$'s payment based on $dr(t_{sj})$;
    for (each item $p_*^{(k)}$ in $QSET(t_{sj})$) {
        Connect $p_*^{(k)}$ to confirm its current availability state;
        Assign the next-lowest payment to $p_*^{(k)}$ to $t_{sj}$ ;
        Update $p_*^{(k)}$ status and execute $t_{sj}$ in a VM on $p_*^{(k)}$ ;
    }
}
```

Second-highest scheduling payment
(Vickrey Auction)

Next-lowest execution payment among nodes
(Reverse Vickrey Auction)

Performance Evaluation

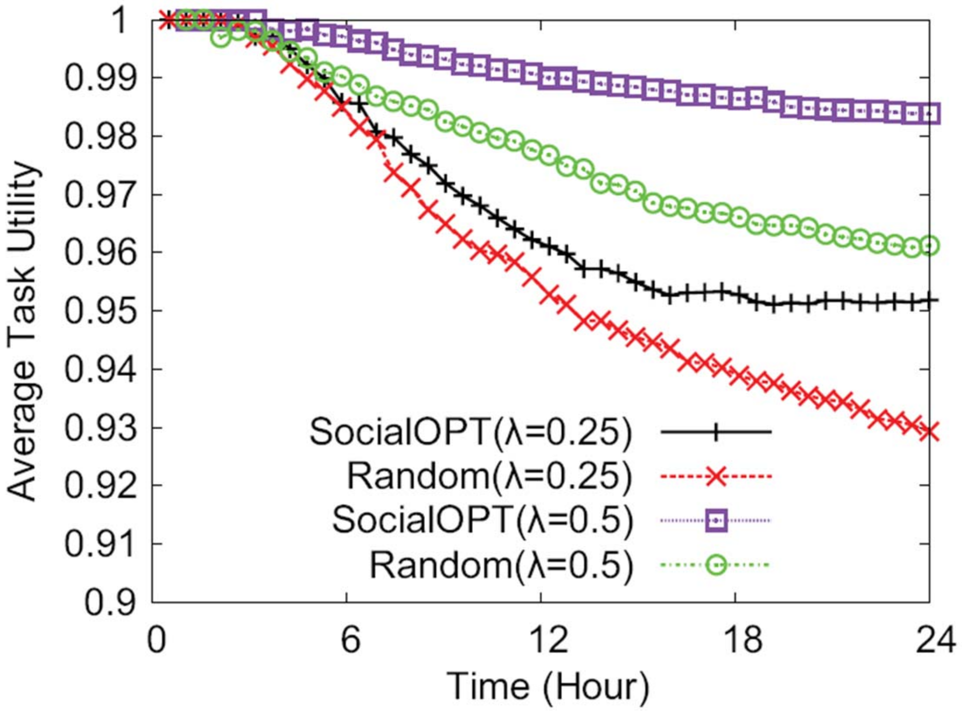
- System Setting
  - PeerSim Tool (even-driven simulation)
    - 4320X100 cycles to simulate 86400 seconds (1 day)
  - 2000 nodes, each is deployed simulated credit scheduler (proportional-share model)
    - # of processors per node: 1,2,4,8
    - Computing capability per processor: 1, 2, 2.4, 3,2 GHz
    - Disk-I/O speed per node: 20, 40, 60, 80 MB/s
    - Memory size per node: 20, 60, 120, 240 GB
    - LAN network bandwidth: 5 ~ 10 Mbps
    - WAN network bandwidth: 0.2 ~ 2 Mps

- Comparision:
  - SocialOPT Selection vs.Non-SocialOPT (Random Selection)

Performance Evaluation

- Average Task Utility (Social Welfare)

$$ATU = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m_i} tu(t_{ij})}{\sum_{i=1}^{n} m_i}$$



Our solution outperform the random node-selection strategy, in different situations with various competitive level of demanded resources

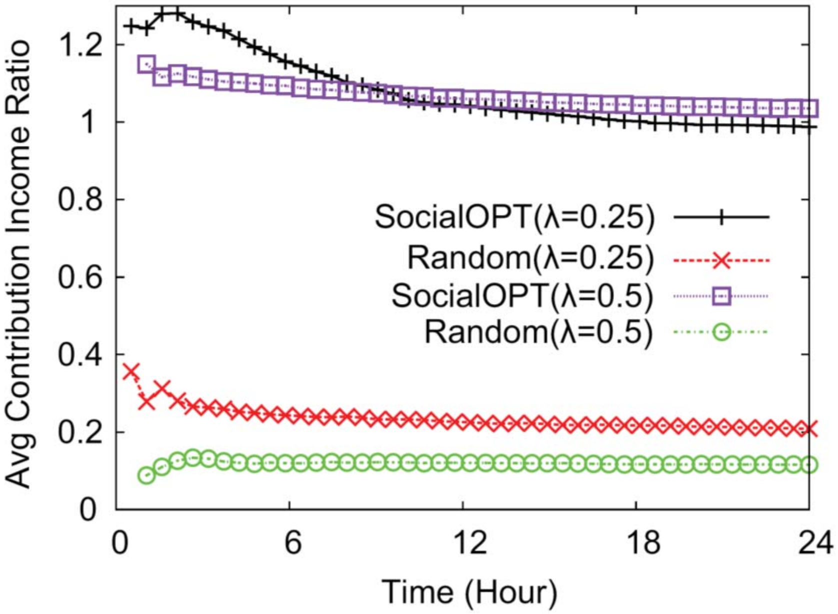
Performance Evaluation

- Average Contribution Income Ratio

$$\frac{1}{n}\sum_{d=1}^{n}\frac{I_{ct}^{real}(p_d)}{I_{ct}^{expect}(p_d)}$$

Real Income of Contributor

Expected Income of Contributor



Under our DVA algorithm, contributors will be much more satisfied with their payoffs.

Performance Evaluation

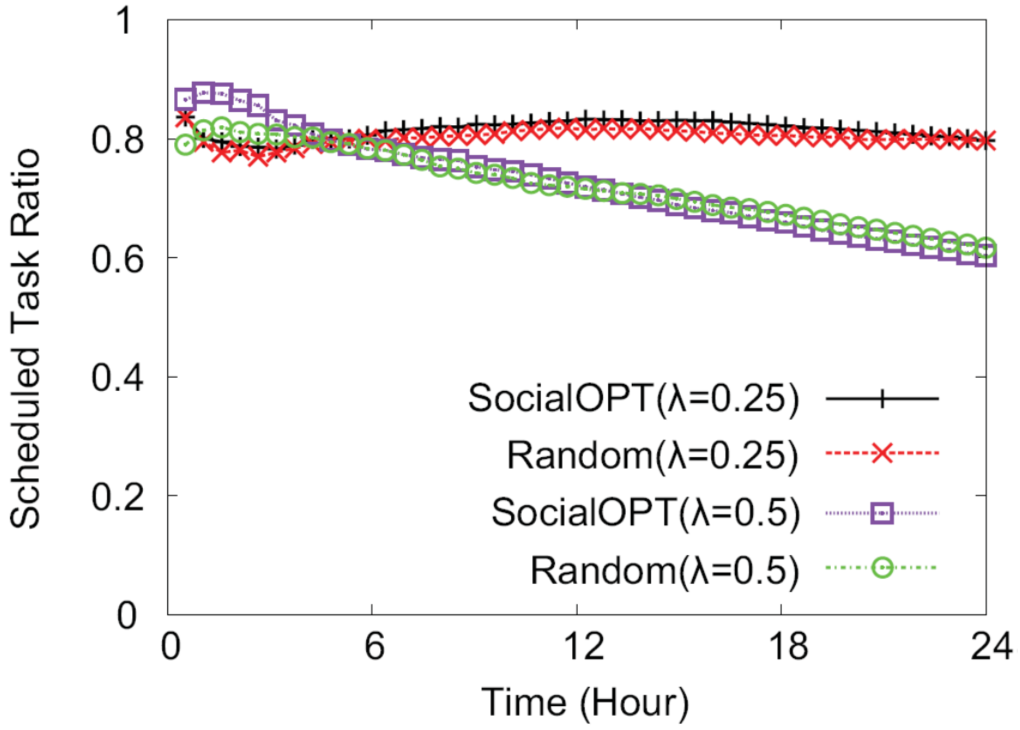
- System Throughput
  - Scheduled Task Ratio $\frac{\text{\# of scheduled tasks}}{\text{\# of total tasks submitted}}$
  - Finished Task Ratio $\frac{\text{\# of finished tasks}}{\text{\# of total tasks submitted}}$
- Fairness Index of Task's Execution Efficiency φ

$$\varphi = \frac{(\sum_{i=1}^{n} \sum_{j=1}^{m_i} e_{ij})^2}{(\sum_{i=1}^{n} m_i) \cdot (\sum_{i=1}^{n} \sum_{j=1}^{m_i} e_{ij}^2)}$$

t_{ij}'s efficiency $e_{ij} = \frac{\text{real execution time}}{\text{estimated value via avg ability}}$

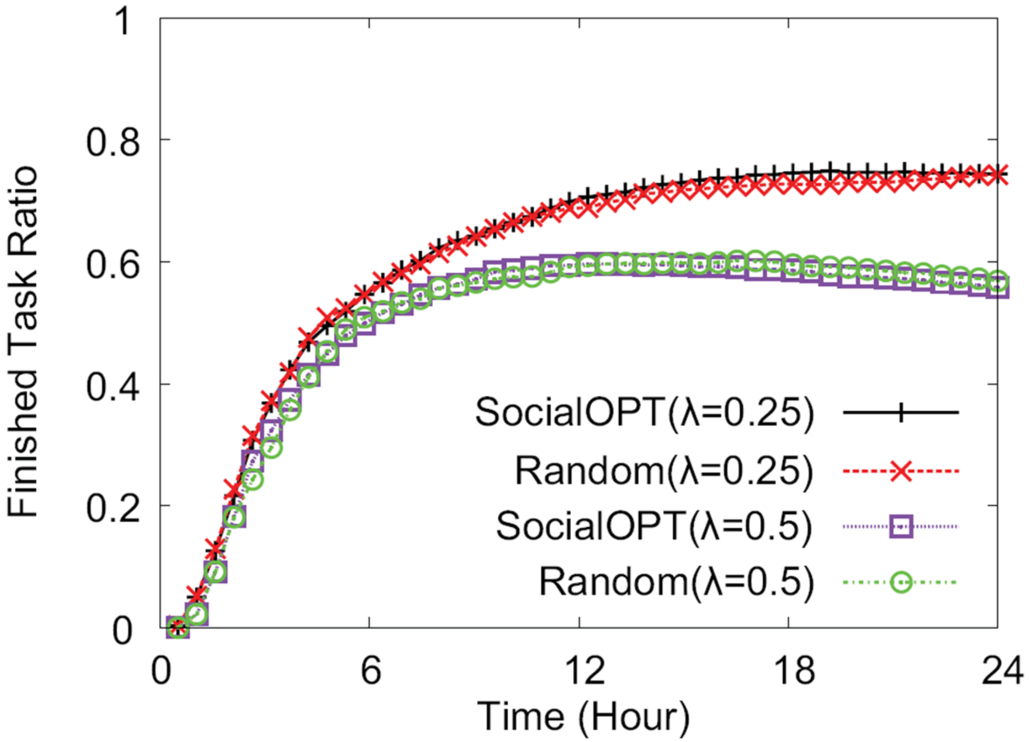
Performance Evaluation

- Scheduled Task Ratio



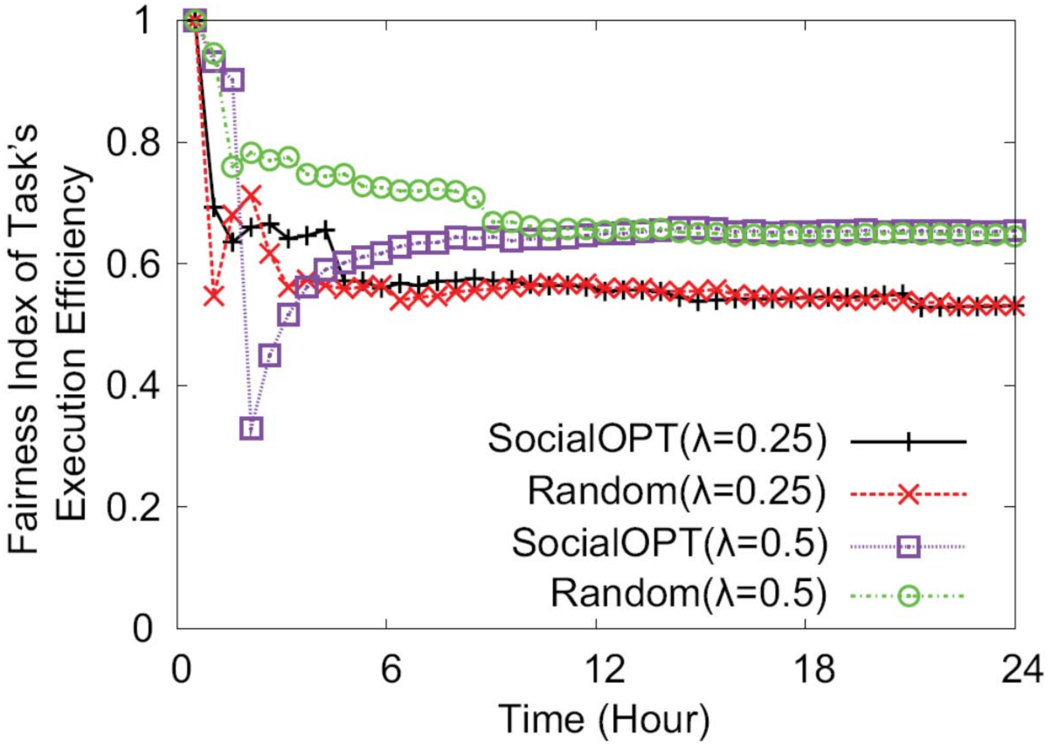
Performance Evaluation

- Finished Task Ratio



Performance Evaluation

- Fairness Index of Task's Execution Efficiency



Conclusion

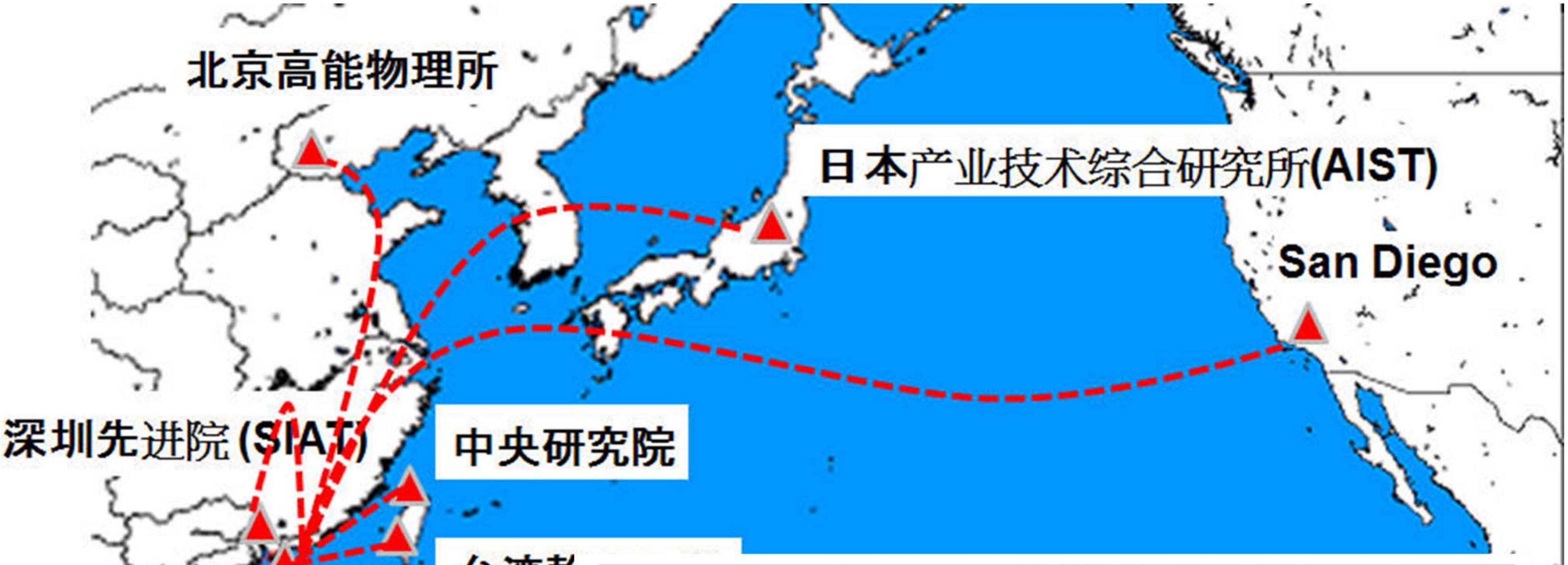
- Contributions of this paper
  - A theoretical one
  - P2P Cloud economy:
    - Design a Dual-Vickrey Auction (DVA) algorithm by considering both sides' payoffs
      - Strategy-proof feature: each participant is willing to reveal its real expectation (e.g. true evaluation of resources)
    - Optimized Social Welfare (Average Task Utility)

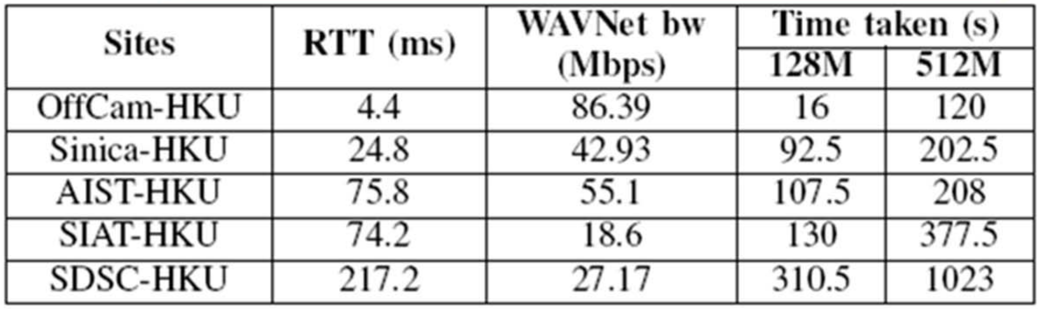
Current status of our project

- P2P desktop Cloud
  - Network Virtualization Layer
    - “WAVNet: Wide-Area Network Virtualization Technique for Virtual Private Cloud”, presented in **ICPP-2011**. (Best Paper Candidate)
  - Virtual Machine Technology
    - “Defeating Network Jitter for Virtual Machines”, presented in **UCC-2011.** (Best Student Paper Award)
  - Resource Discovery Protocol
    - “Probabilistic Best-fit Multi-dimensional Range Query in Self-Organizing Cloud”, presented in **ICPP-2011.**
    - “Decentralized Proactive Resource Allocation for Maximizing Throughput of P2P Grid”, appears in **JPDC-2011**.
  - **The Economy of P2P Desktop Cloud**
    - “Social-optimized Win-win Resource Allocation for Self-organizing Cloud”, presented in CSC-2011.

Network virtualization Layer

- **WAVNet: Wide-Area Network Virtualization Technique for Virtual Private Cloud (ICPP 2011, Best Paper Candidate)**





Sites	RTT (ms)	WAVNet bw (Mbps)	Time taken (s)	
			128M	512M
OffCam-HKU	4.4	86.39	16	120
Sinica-HKU	24.8	42.93	92.5	202.5
AIST-HKU	75.8	55.1	107.5	208
SIAT-HKU	74.2	18.6	130	377.5
SDSC-HKU	217.2	27.17	310.5	1023

Future Work

- Improve fault-tolerance by combining the replica-task execution strategy.
  - Peer nodes dynamically join and leave without control.
- To build a more robust system, we should encourage volunteers to stay as long as possible.
  - Longer stay → more credit; Shorter stay → less credit
- The model is still a bit simple..
  - Some assumptions are too strong
- Implement the algorithm into our real platform, and get experimental data from the real world.
  - Currently, we only consider sequential tasks. How about other types of applications?

Thank You!
Q & A

Backup slides

More Related Work

- Social-optimized **Win-win** Resource Allocation Model (Economy-based Resource Allocation)
  - Reciprocation-Based Economy (RBE)
    - Each node always donates its service to others based solely on the record of their past bilateral inverted service actions. So, the nodes who contribute more will get more in return when they make requests.
    - Not flexible since there are no currencies to coordinate interests between two sides.
    - OurGrid and SHARP
  - Nash Bargaining Solution (NBS)
    - Any requester must make an agreement on a price by negotiating with another supplier before the consumption.
    - Although NBS may ensure demand and supply match reciprocally, the matching procedure is relatively low-efficient because the tasks cannot be started/executed without a couple of bargaining rounds.
    - Nimrod-G and MobileGrid
  - Auction-Based Solution (ABS)
    - Google's Planet Cluster, SCDA, etc.