Incentives and General Welfare Functions in the Off-Line Cluster Scheduling Problem

Orna Agmon

agmon@tx.technion.ac.il

The research thesis was done under the supervision of Dr. Rann Smorodinsky in the department of industrial engineering.

Agenda

- The off line cluster scheduling environment
- The game
- Mechanism properties
- The proposed class of mechanisms
- Quality of proposed mechanisms
- Related work

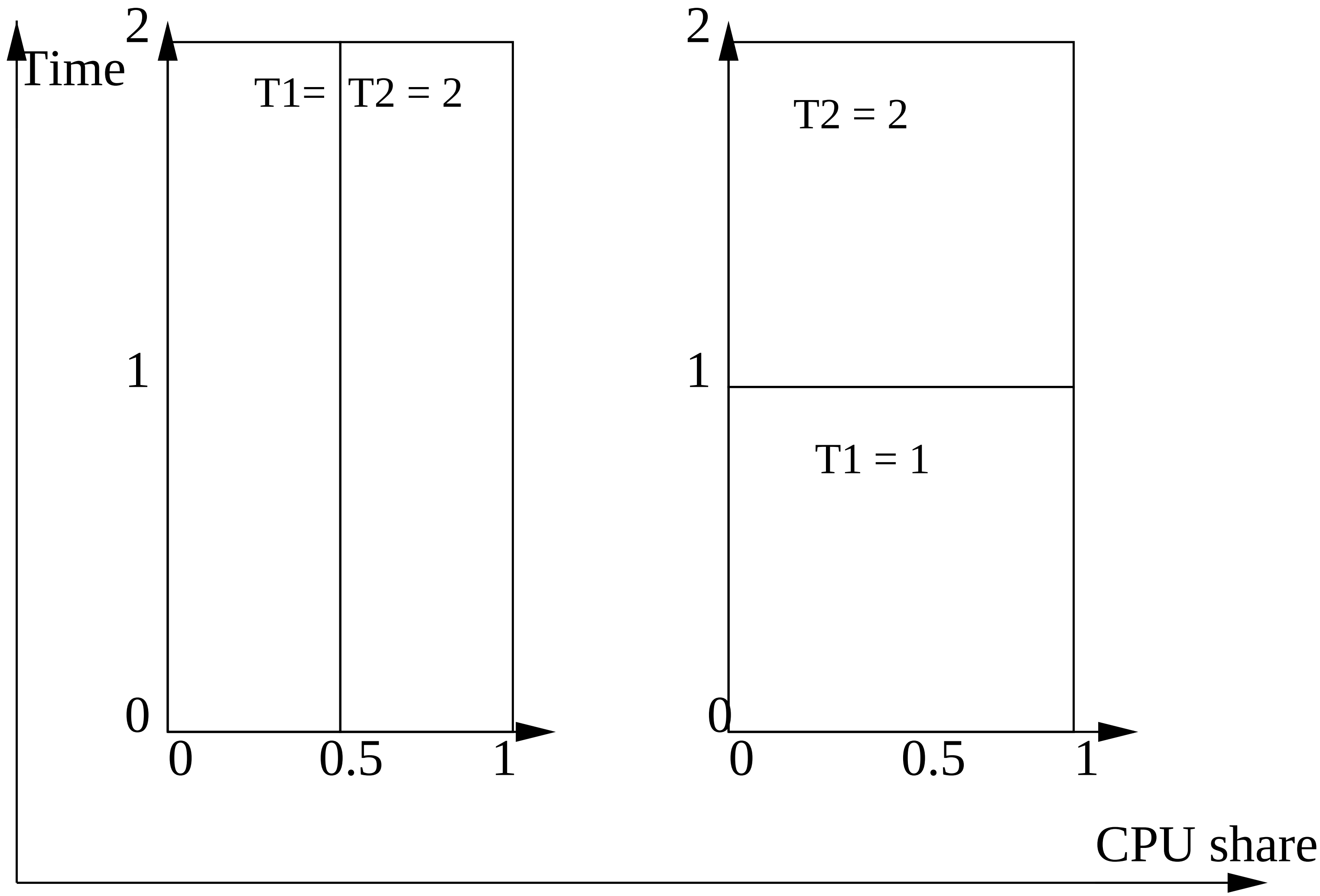
The Off-Line Cluster Scheduling Environment

- N selfish agents with jobs of lengths $\theta_1 \ldots \theta_N$ (private information).
- M single CPUs with computing power $c_1 \ldots c_M$: the **cluster**.
- An institution, which owns the cluster.

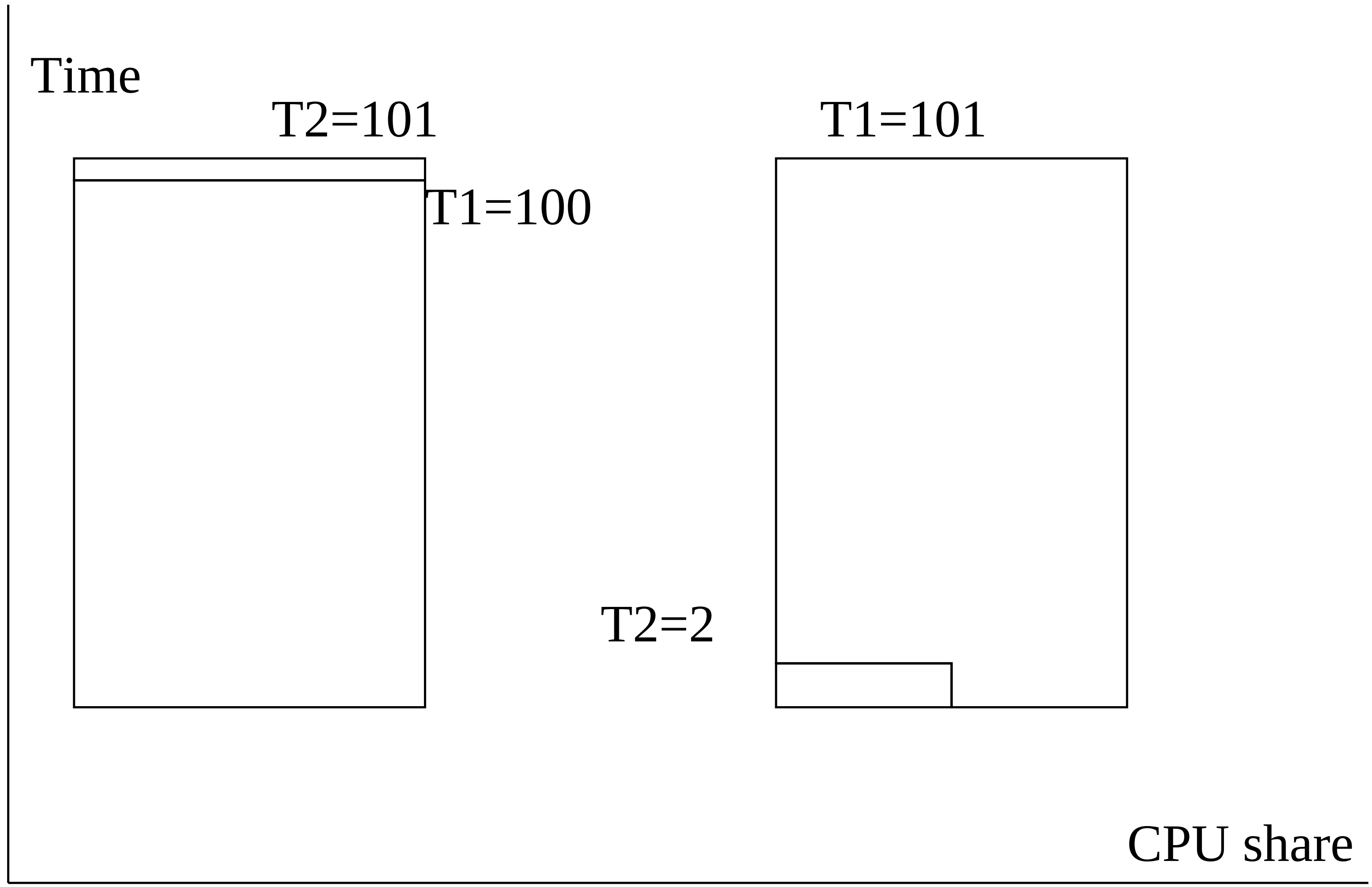
Utilities

- An agent's utility $U_n = X_n - T_n - P_n$:
  1. $X_n > 0$ Agent's value for executing the job - common knowledge.
  2. $T_n > 0$ The output time: time in which the agent receives the output
  3. P_n Price the agent pays to the institution
- The institution's utility (the "social welfare") is a **general** function of $\vec{T}, \vec{P}$.

Why does the institution need a central scheduler?



Sequential access is not good enough.

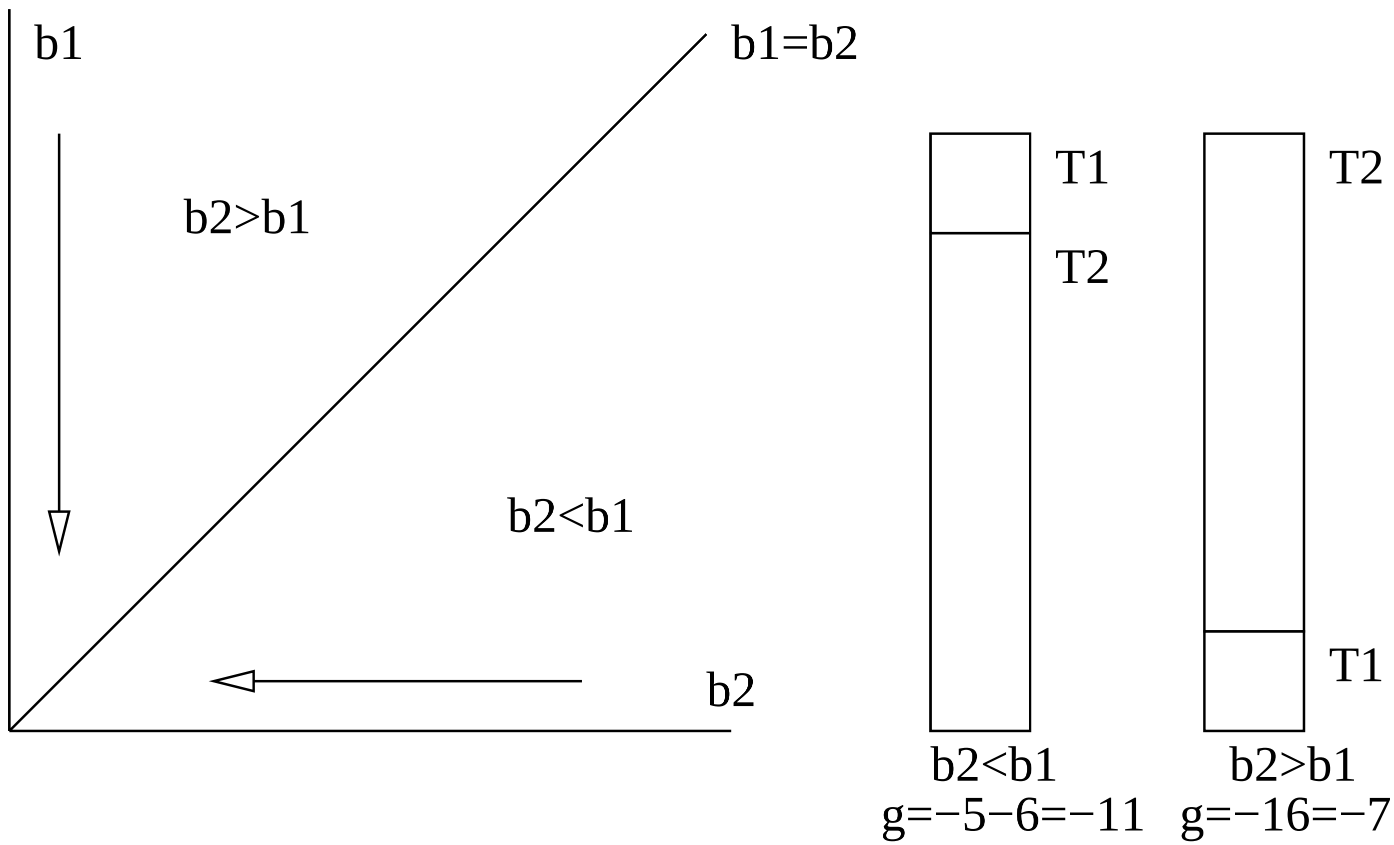


What is a straightforward scheduling mechanism?

1. The agents declare the lengths of their jobs $\vec{b}$.
2. The institution divides the jobs among the CPUs.
3. The institution sets the order of execution within each CPU.

Incentives in the straightforward mechanism

Example: one CPU, the institution maximizes $g_{\sum} = -\sum_{k=1}^{N} T_k$:



The Off-Line Cluster Scheduling Mechanism

The mechanism supplies:

1. $\vec{P}$: Prices.
2. $\vec{T}$: Output times.
3. A : A static allocation.
4. Job control tools: operators on a status $Q = (A, \vec{T})$

An allocation

An **allocation** A of a set of jobs $\mathcal{N}$ is composed of:

1. **Partition to disjoint subsets**: $\forall m \in \mathcal{M}$, $\mathcal{N}_m^A \subset \mathcal{N}$ is a subset of $\mathcal{N}$ s.t.

$$\cup_{m \in \mathcal{M}} \mathcal{N}_m^A = \mathcal{N}$$
$$\forall m \neq k \quad \mathcal{N}_m^A \cap \mathcal{N}_k^A = \emptyset$$

2. **Work functions**: $\forall m \in \mathcal{M}, \forall n \in \mathcal{N}_m^A$, $X_n^A(t) : \mathbb{R}_+ \mapsto [0,1]$, a continuous to the right function, denotes the percentage of CPU m which is devoted to job n at time t, and satisfies $\sup\{t : X_n^A(t) > 0\} < \infty$, as well as $\forall m \in \mathcal{M}$

$$usage: \quad X^{A,m} := \sum_{n \in \mathcal{N}_m^A} X_n^A \leq 1$$

Job control tools

- *Early* - release the output earlier than planned.
- *Renice* - let the job finish the required work by continuing to use only a share s_{renice} of the CPU.
- *Postpone* - let the job finish the required work at a later time, no sooner than s_{post} after its original ending time.
- *Close* - close a (full) gap in the usage.

In real life, not all tools are available on every system.

Times

- At time E_n^A the cluster stops executing job n, under allocation A.
- At time L_n, job n is done.
- At time T_n output of job n is given to agent n.

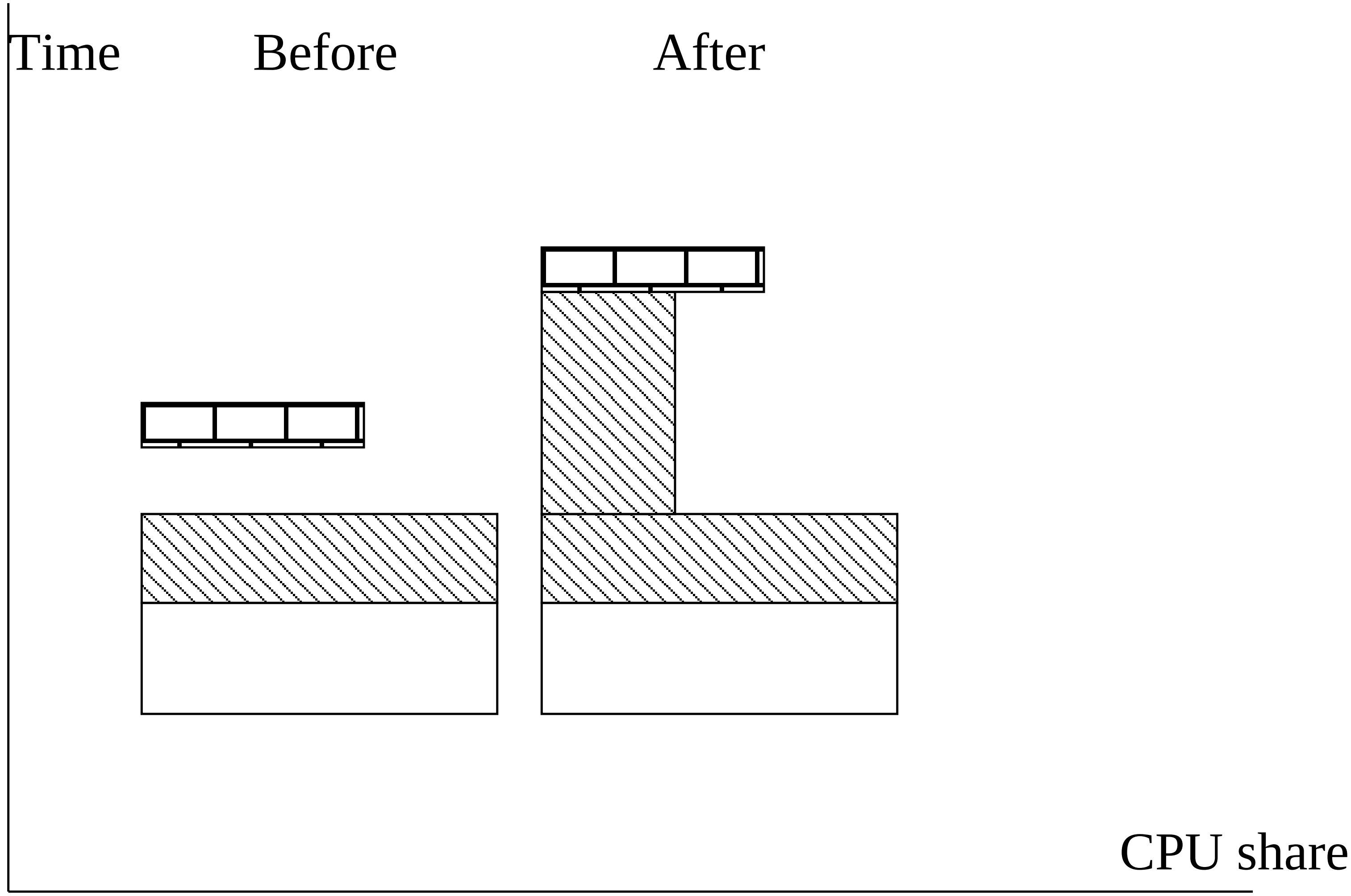
Early

$\forall n \in \mathcal{N}$, we define an operation $EARLY_n$, **early release of job n** on a status Q, as follows:

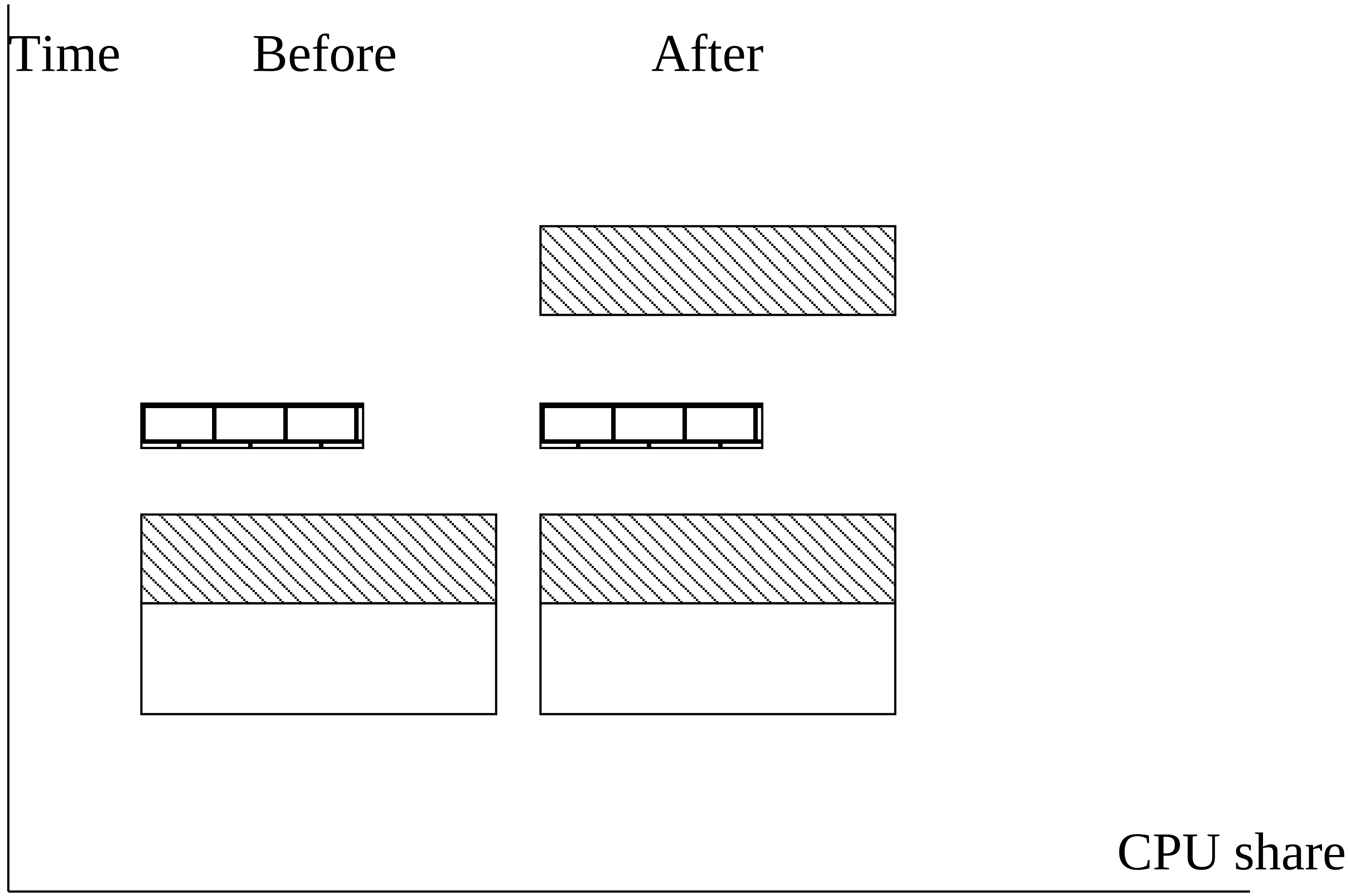
$$T_l^{EARLY_n(Q)} = \begin{cases} T_l^Q & l \neq n \\ min(E_n^A, L_n) & l = n. \end{cases}$$

$$A^{EARLY_n(Q)} = A$$

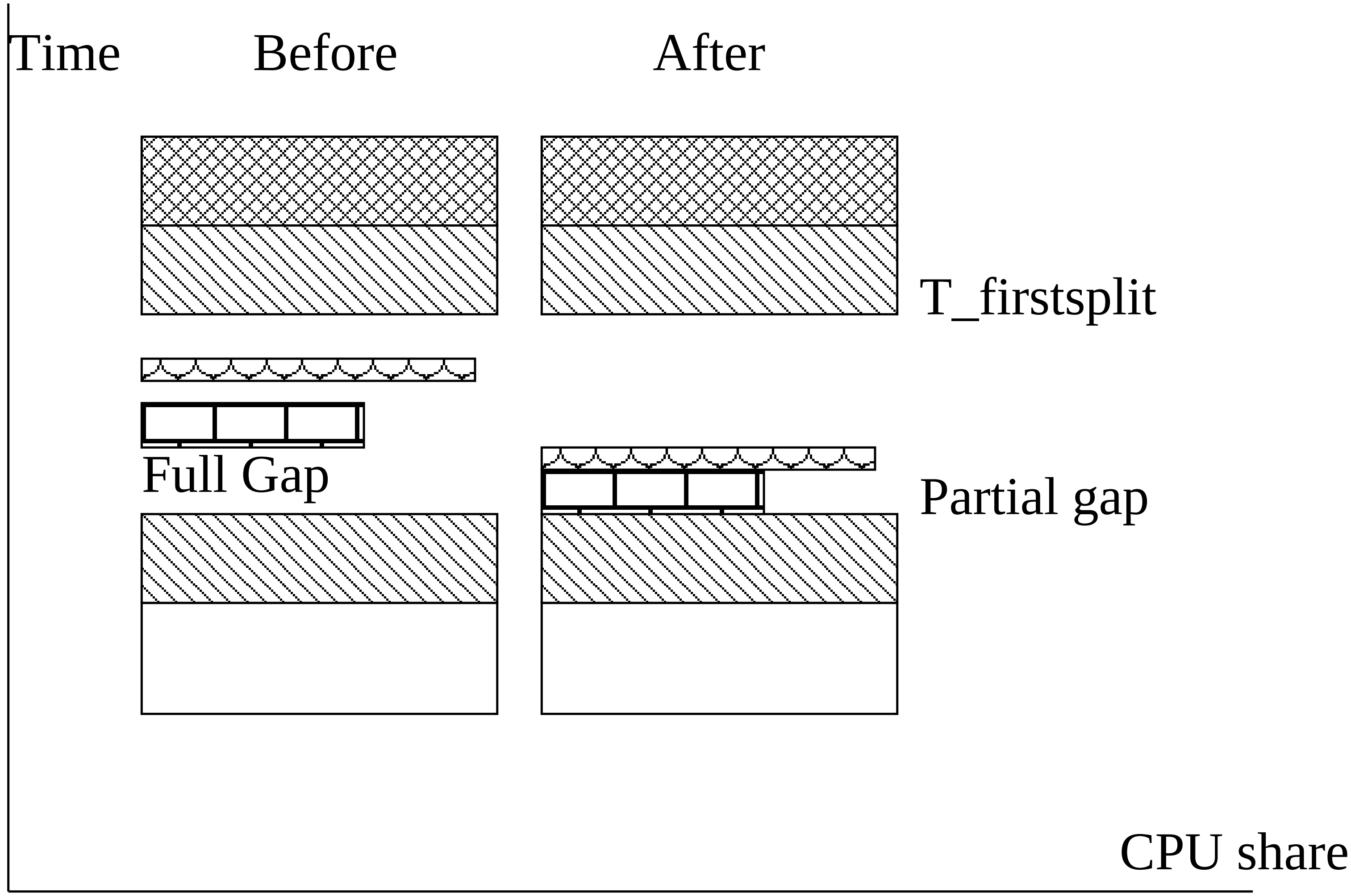
Renice



Postpone



Close


Time
Before
After
T_firstsplit
Full Gap
Partial gap
CPU share

$$\left.\begin{array}{l}\text{Agents}\\ \text{An Institution}\\ \text{A Mechanism}\\ \text{Utilities}\end{array}\right\} \Rightarrow \text{A Game}$$

The game stages

- Institution's commitment to a mechanism:
  1. $A(\vec{b})$
  2. $\vec{T}(\vec{b})$
  3. $P(\vec{b})$
  4. the available job control tools and their triggers.
- Declaration: The agents declare a (possibly true) job length $\vec{b}$.

The game stages (2)

- Realization: according to the initial commitment and $\vec{b}$, the institution decides on:
  - Initial allocation (which maximizes g) and output times $Q = (A, \vec{E}^A)$.
  - Prices $\vec{P}$.
  - the job control tool parameters: s_{post}, s_{renice},.
- Payment: agents pay $\vec{P}$.
- Execution (According to $Q+$ job control tools).

Mechanism Properties (1)

Incentive compatibility (IC)

1. Truth telling is a dominant strategy .
2. Truth telling is in ex-post equilibrium.
   A strategy $S : \Theta \mapsto \Theta$ is in **Ex-Post equilibrium** if it is the best strategy against agents using the same strategy, regardless of what their lengths are: $\forall n \in \mathcal{N}, b_n \in \Theta, \vec{\theta} \in \Theta^N$,

$$U_n(S(\theta), \vec{\theta}) \geq U_n((b_n, S_{-n}(\theta_{-n}), \vec{\theta})).$$

Dominant strategies $\Rightarrow$ ex-post eq. $\Rightarrow$ Bayes-Nash eq.

Mechanism Properties (2)

- Budget considerations:
  1. A balanced budget.
  2. A positive rent.
- Safety margins:
  1. Two sided.
  2. One sided.

Mechanism Properties (3)

- Prices depend on declaration only.
- Justness.
- Social welfare of the final status, given $\vec{\theta}$.
- Unlimited input (scalability).

We shall now focus on the $g_{\sum}$ function as an example:

Light VCG mechanism

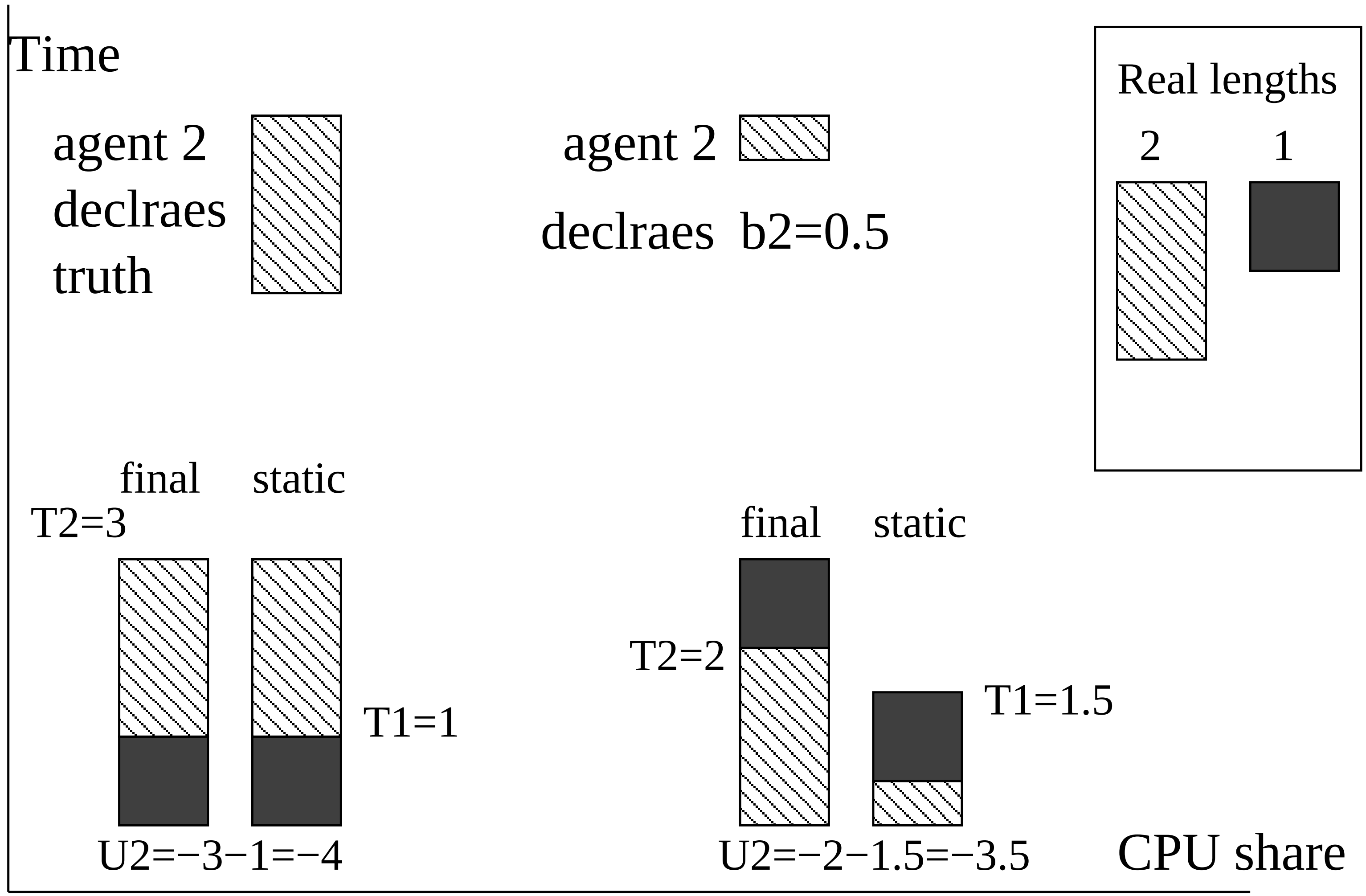
The VCG mechanism is known to implement the $g_{\sum}$ social welfare function in many environments. In the off-line cluster scheduling environment, we could have:

- Vickrey-Clarke-Groves (VCG) prices

$$P_n = \sum_{k \neq n} T_k(\vec{b}, \vec{b}).$$

- The institution optimizes $g_{\sum}$.
- *Renice* is the only job control tool: $RN_n(E_n^A, 1, \theta_n)$ if $\theta_n > b_n$.

Light VCG is not IC



Setting VCG payments is not enough.

How can the institution give the agents an incentive to tell the truth?

Harsh Mechanism

- $\vec{P} = \vec{0}$.
- The only job control tool is *Postpone*: $b_n \neq \theta_n \Rightarrow POST_n(min(E_n^A, L_n), \infty, \theta_n + \epsilon)$.

This mechanism is:

- just,
- scalable,
- budget balanced and with prices known in advance.

However,

- no safety margins,
- the worst social welfare when agents lie.

Can we do better?

The *Renice* and *Postpone* mechanisms

- Job control triggering:
  - $b_n < \theta_n$ triggers *Renice* or *Postpone*,
  - $b_n > \theta_n$ triggers *Close*,
  - $b_n = \theta_n$ triggers *Early*.
- VCG Payments.
- A optimizes $g_{\sum}$.

Positive Results for $g_{\sum}$

- In a system with *Postpone*, *Close* and *Early*:
  - It is possible to implement $g_{\sum}$, the sum of utilities function, in dominant strategies.
  - s_{post} poses a limit on the tolerated lie.
- In a system with *Renice*, *Close* and *Early*:
  - It is possible to implement $g_{\sum}$ in ex-post equilibrium.
  - s_{renice} can take a certain range of values, but does not limit the input nor the lie tolerance.

Can these mechanisms be extended in order to implement a general social welfare function?

An extension to a general g

- $A = o(\vec{b}, \vec{b})$ optimizes a general social welfare function g.
- Same job control as the *Renice* and *Postpone* mechanisms.
- Extended VCG payments (EVCG):

$$\begin{aligned} P_n(\vec{b}) &= -T_n(\vec{b}, \vec{b}) + T_{\Sigma,n}(\vec{b}, \vec{b}) + \sum_{k \neq n} T_{\Sigma,k}(\vec{b}, \vec{b}) = \\ &= COMPENSATION + \ VCG \ \ PAYMENT \\ &= -T_n(\vec{b}, \vec{b}) + \sum_{k=1}^{N} T_{\Sigma,k}(\vec{b}, \vec{b}). \end{aligned}$$

Positive Results:

The EVCG mechanism has the same results for a general social welfare function as the VCG mechanism for the $g_{\sum}$ function!

Discussion

- input limitations
- safety margins
- individual rationality
- justness
- social welfare when agents lie
- computability and off line calculations

Discussion (2): budget considerations

- Rent is no necessarily positive (example: $g = -\sum (T_k - T_0)^2$)
- Regular social welfare function $\Rightarrow$ non-negative rent.

Scheduling games

References

[1] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[2] Michael P. Wellman, William E. Walsh, Peter R. Wurman, and Jeffrey K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35(1):271–303, 2001.

Summary

In the off-line cluster scheduling environment, it is possible to implement a general social welfare function. We devised two variants of a mechanism, in which truth telling is a preferred strategy.

1. *Postpone* based- just, upper safety margin, limited input. Implementation in dominant strategies.
2. *Renice* based- unjust, safety margins, unlimited input. Implementation in ex-post equilibrium.

Game theory

References

[1] R. Holzman, N. Kfir-Dahav, D. Monderer, and M. Tennenholtz. Bundling equilibrium in combinatorial auctions. mimeo, Technion, http://iew3.technion.ac.il/~moshet/rndm11.ps, 2001.

Individual equilibrium and learning in process sharing systems. *Operations Research*, 46:776–784, Dec. 1998.

[2] Ann Van Ackere. Conflicting interests in the timing of jobs. *Management Science*, 36(8), 1990.

Scheduling

References

[1] Jahanzeb Sherwani, Nosheen Ali, Nausheen Lotia, Zahra Hayat, and Rajkumar Buyya. Libra: An economy driven job scheduling system for clusters. Technical report, The University of Melbourne, July 2002. http://www.cs.mu.oz.au/~raj/grids/papers/libra.pdf.

[2] Jianzhong Du and Joseph Y.-T. Leung. Minimizing total tardiness on one machine is NP-hard. *Mathematics of Operations Research*, 15:483–494, 1990.

[3] A. Barak, S. Guday, and R. Wheeler. *The MOSIX Distributed Operating System, Load Balancing for UNIX*, volume 672.

Springer-Verlag, 1993. http://www.mosix.org.

[4] Ulrik Kjems. Jobd. http://bond.imm.dtu.dk/jobd/.

[5] Miron Livny et al. Condor - high throughput computing. http://www.cs.wisc.edu/condor/.

[6] Rajkumar Buyya, David Abramson, and Jonathan Giddy. Nimrod/g: An architecture for a resource management and scheduling system in a global computational grid. In *The 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000)*. IEEE Computer Society Press, USA, May 2000.

[7] Babak Falsafi and Maurio Lauria, editors. *REXEC: A Decentralized, Secure Remote Execution Environment for Clusters.*, volume 1797 of *Lecture Notes in Computer Science*. Springer, 2000.

Questions???

Extra Slides

Postpone

$\forall s, r \in \mathbb{R}_+, \forall n \in \mathcal{N}$, we define an operation $POST_n(r, s, \theta_n)$, **postpone job n from time r to time s and let it continue until it performs a total work of** θ_n on a status Q as follows:

$$POST_n(r, s, \theta_n)(Q) = RP_n(0, L_n) \circ$$
$$CLOSE_m(r, s) \circ ES_n(\infty) \circ GAP_n(r, s)(Q)$$

where $n \in \mathcal{N}_m^A$, and L_n is such that

$$c_m \int_{t=0}^{L_n} X_n^{CLOSE_m(r,s) \circ ES_n(L_n) \circ GAP_n(r,s)(Q)} dt = \theta_n.$$