GRAAP-WG Meeting, GGF17
10-12 May 2006
Tokyo, Japan

Dynamic Agreement:
A Function Based Approach to WS-Agreement

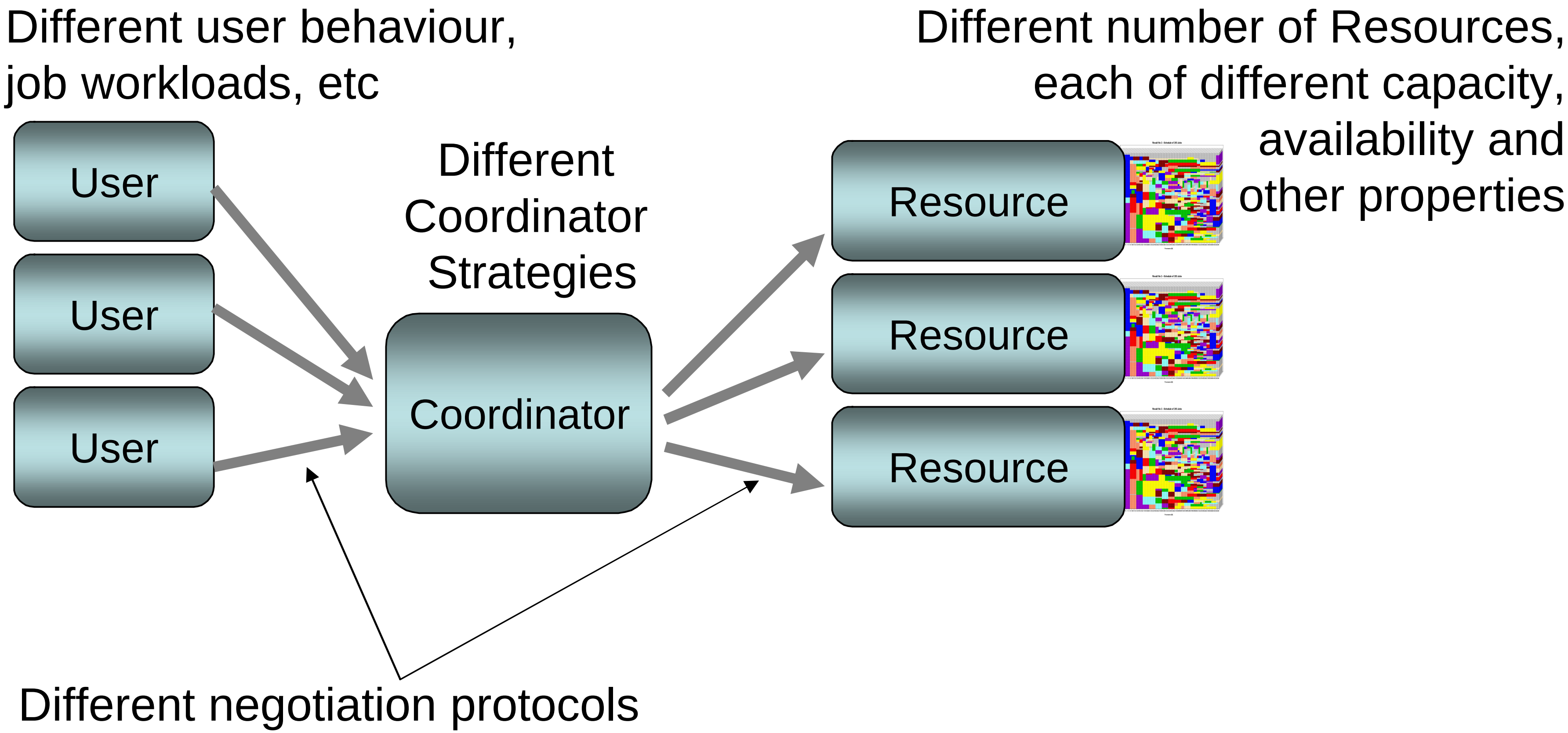
Based on publication by **Rizos Sakellariou** and **Viktor Yarmolenko**, *On the Flexibility of WS-Agreement for Job Submission*, Proceedings of the 3rd International Workshop on Middleware for Grid Computing MGC '05, vol. 117, 1-6 (Nov. 2005)

Engineering and Physical Sciences Research Council

e-mail: viktor.yarmolenko@manchester.ac.uk
http://www.gridscheduling.org

Background

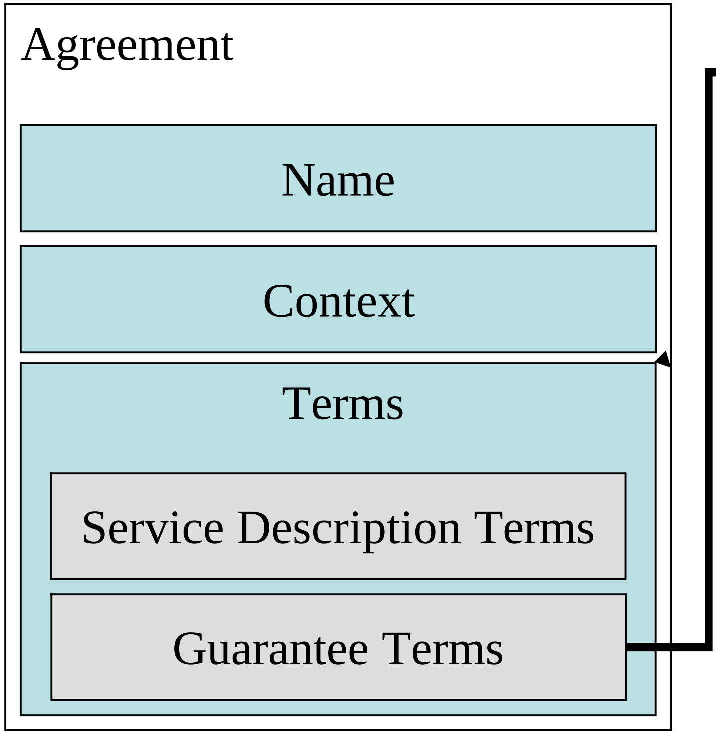
Parallel Job submission using SLAs



And many more (such as pricing policies, system topology, external factors) all in the context of Service Level Agreements.

Introduction

Our current interest in WS-Agreement

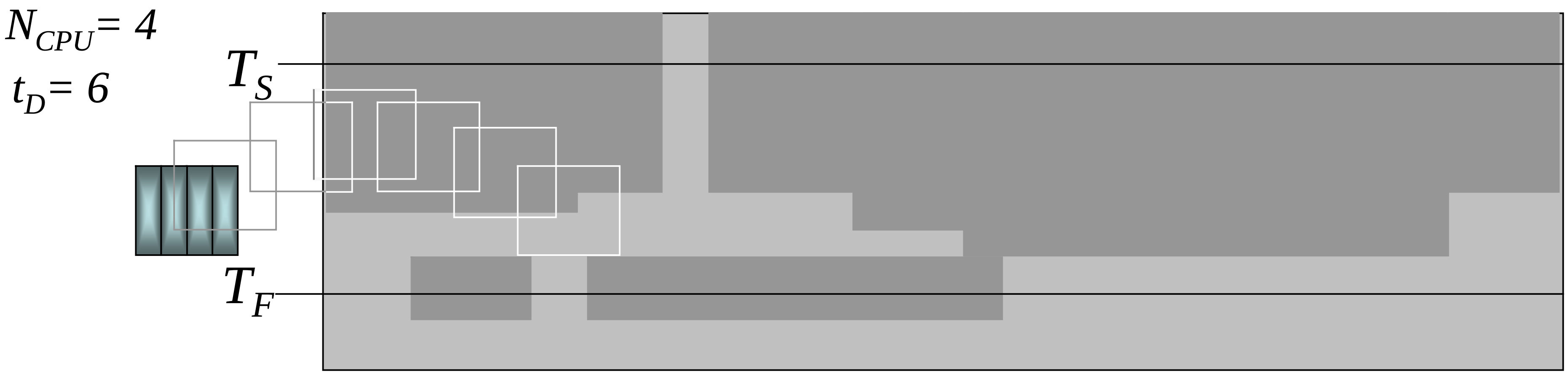


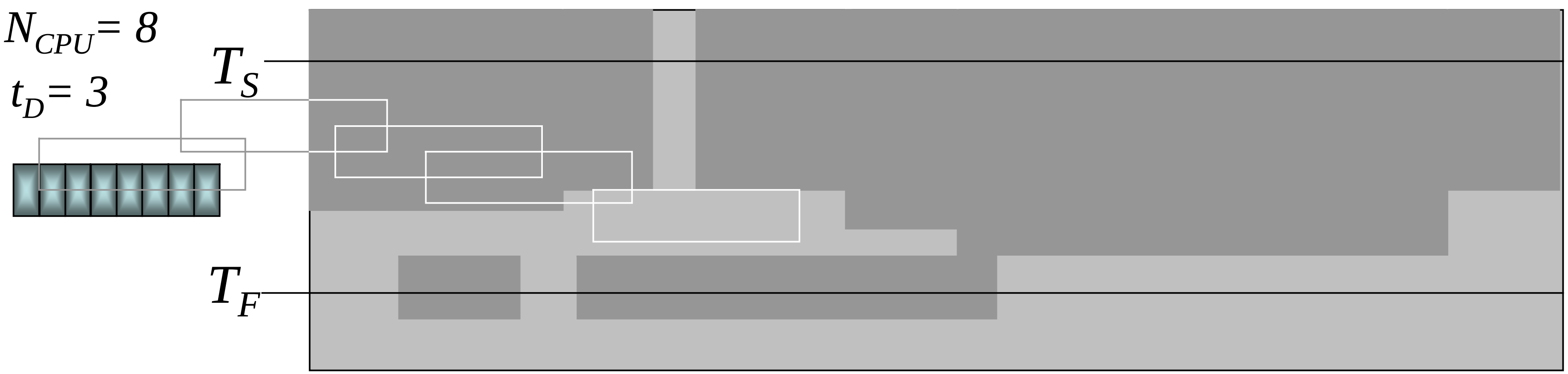
Service Level Objectives Business Value List
Service Level Indicators

SLO	SLI	BVL
CPUs	8	5
Start time >	9:00	10
Finish time <	23:00	4
Job Duration <	5	1
Job done:		59
Price per time unit:		8
Penalty for delays:		1

Motivation

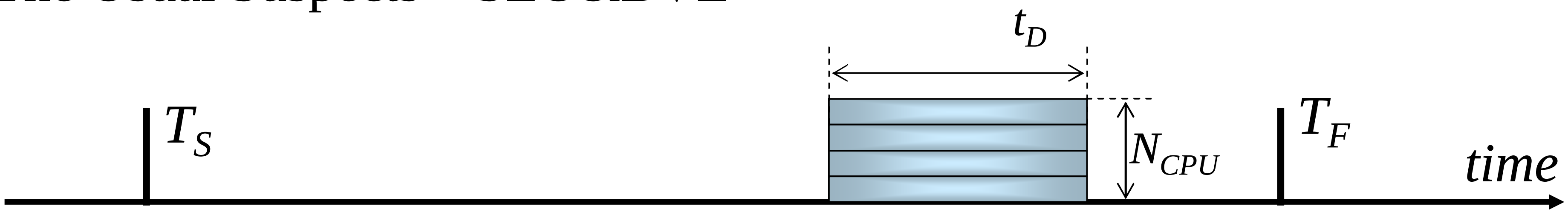
Example with parallel job scheduling





Terms of Agreement

The Usual Suspects – SLO&BVL



SLO: T_S – the earliest time the Job is allowed to start

SLO: T_F – the latest time the Job is allowed to finish

SLO: N_{CPU} – number of CPU nodes required for the Job

SLO: t_D – projected Job duration time for N_{CPU} nodes

SLO: t_{UP} – *uniprocessor Job duration time (CPU-hours)*

BVL: V_{pr} – *the price for executing the Job*

BVL: V_{pn} – *the penalty for failing the Job*

BVL: V_{tot} – *final value of the agreement (optional)*

Extending Terms of Agreement

More Flexibility!!!

A list of universal variables

$x\ y\ z\ h$

$\Delta\ \pi\ \alpha\ \beta$

A list of predefined common functions

Possibility to describe agreement terms as functions

Universal Variables and Functions

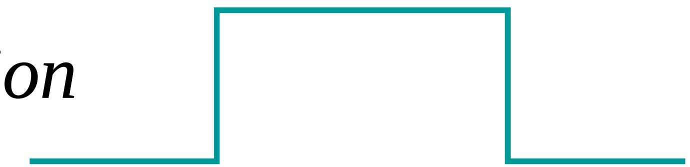
Universal Terms – Useful Variables & Functions

UT: t_{curr} – *current wall clock time*

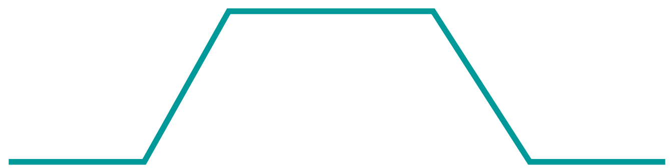
UT: $R_{ld}(t_{curr})$ – *Resource load @ time: current or any other*

UT: t_S – *actual Job execution start time*

UT: $f_{norm}(t, low, high)$ – *binary function*



UT: f_{tr}*(t ,low,a ,high ,ß)* – *trapezium*



Guarantee Terms as Functions

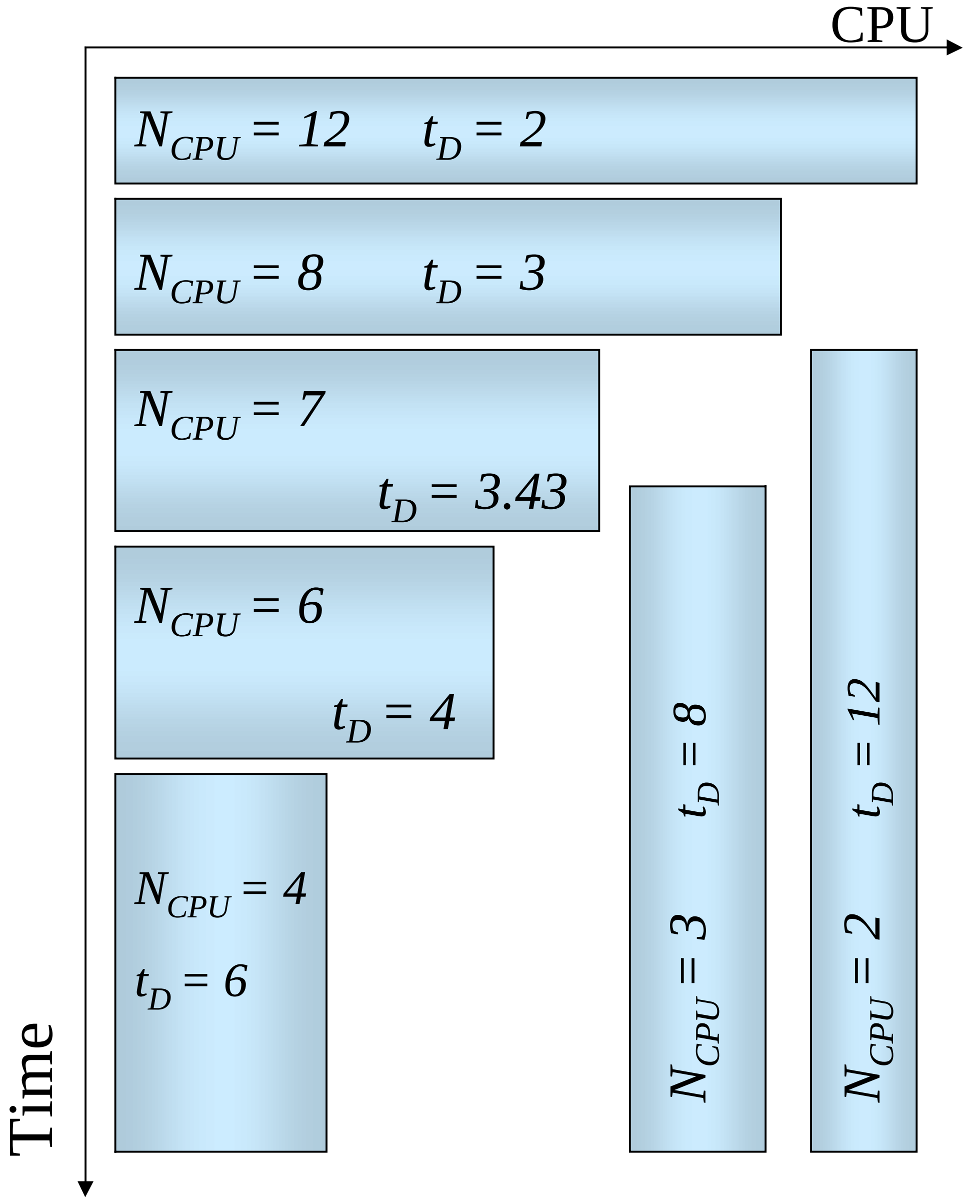
Variable Number of CPUs per Job

SLO: $N_{CPU} = \{2,3,4,..\}$

SLO: $t_D = \dfrac{t_{UP}}{N_{CPU}}$

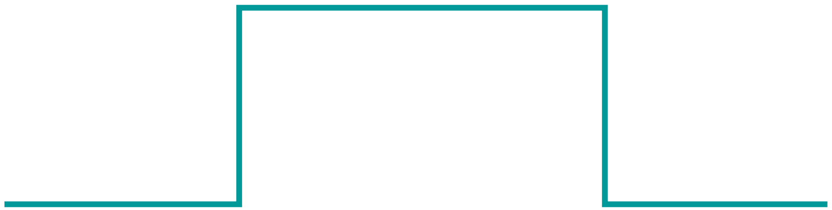
SLO: $t_{UP} = 24$

SLO: $X_{other} = const$



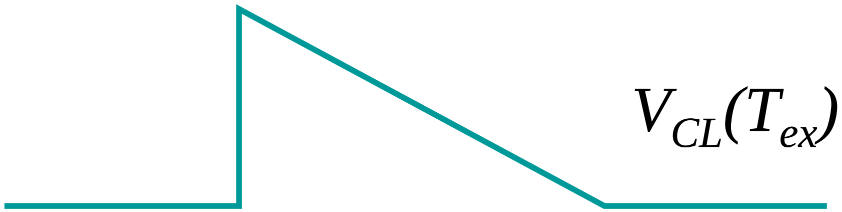
From simple pricing to time dependant

Standard: willing to pay a fixed amount as long as the job starts and finishes within T_S and T_F



$$V_{CL}(T_{ex}) = \begin{cases} 1, & T_S = T_{ex} = (T_F - t_D) \\ 0, & T_S > T_{ex} > (T_F - t_D) \end{cases}$$

ASAP: willing to pay a higher rate if job starts earlier, but still within T_S and T_F times



$$V_{CL}(T_{ex}) = \begin{cases} \left(1 - \frac{T_{ex} - T_S}{T_F - t_D - T_S}\right), & T_S = T_{ex} = (T_F - t_D) \\ 0, & T_S > T_{ex} > (T_F - t_D) \end{cases}$$

Guarantee Terms as Functions

Defining the Value of the Service

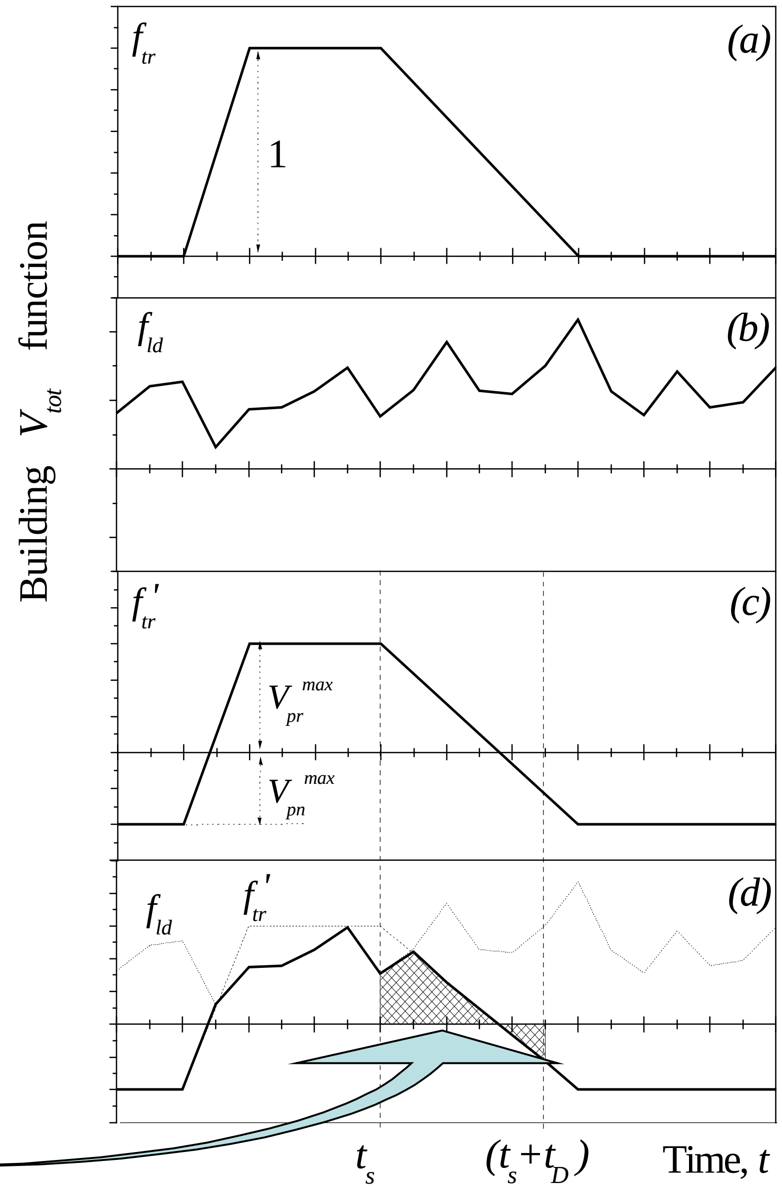
UT: t_{curr}

UT: $R_{ld}(t_{curr}) = f_{ld}$

SLO: $t_D = \dfrac{t_{UP}}{N_{CPU}}$

SLO: $X_{other} = const$

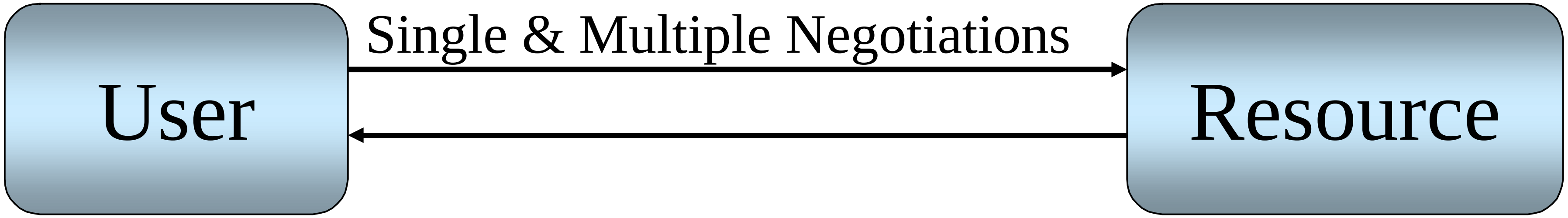
BVL: $V_{tot} = f(R_{ld}, t_s, N_{CPU}, \ldots)$



Suddenly life becomes more interesting

Experiment

The Model



Set of ~800 Job requests, for which a solution exists where the 100% utilisation is possible on Resource (600 hours x 64 CPUs).
Generated 10 independent sets

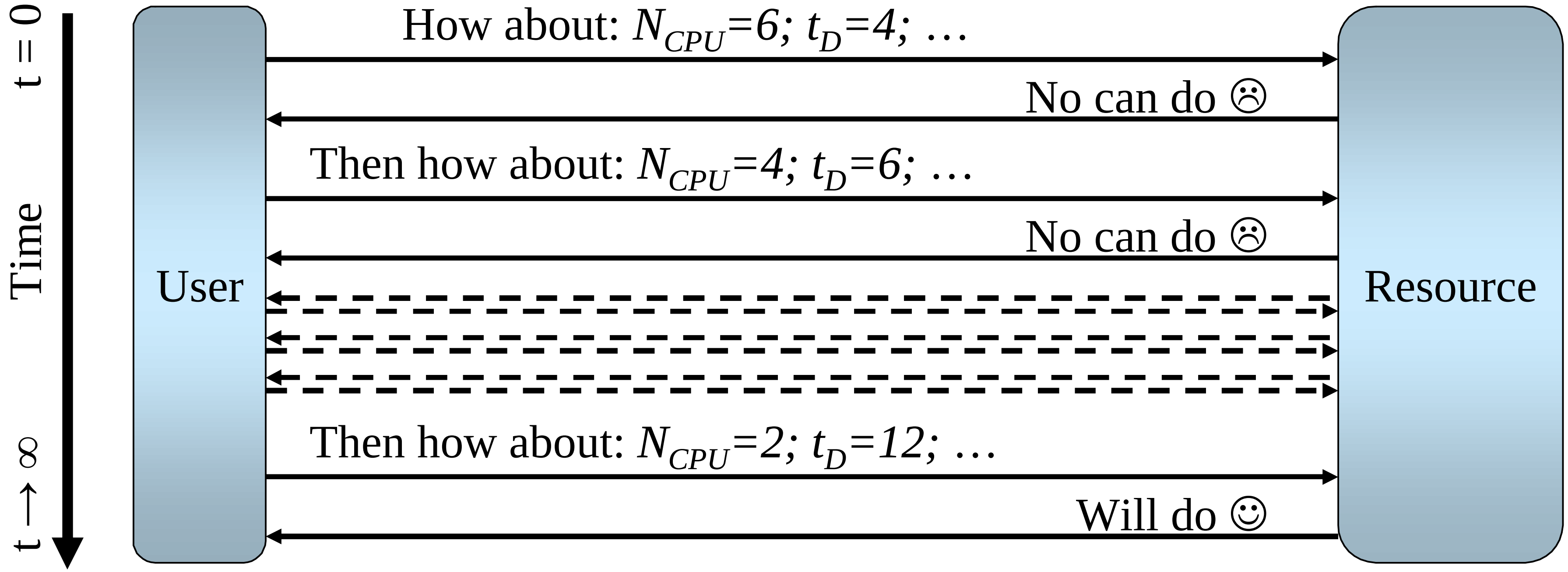
Capacity of 64 CPUs and available for 600 hours

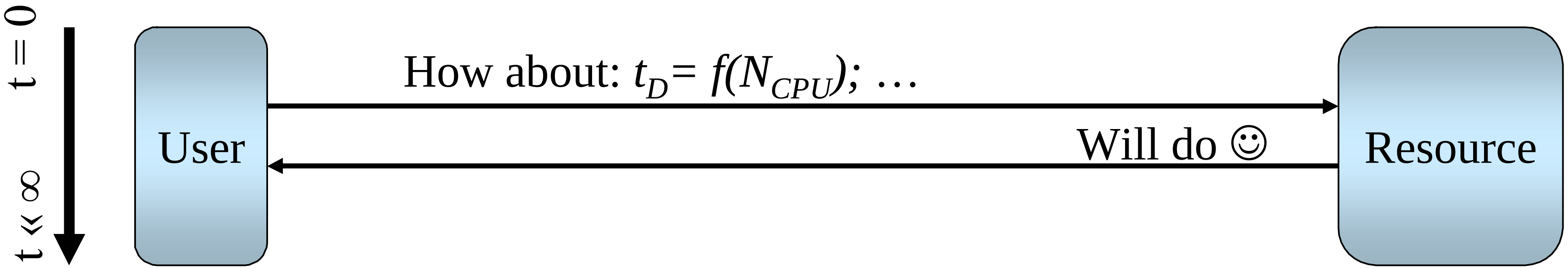
Scheduling by the earliest deadline first (single iteration)

$$\langle t_D \times N_{CPU} \rangle = 21.85$$

Experiment

Variable CPU Scenario (Original vs. Expressive SLA)





Experiment

Only Single Negotiation is Allowed

Experiment

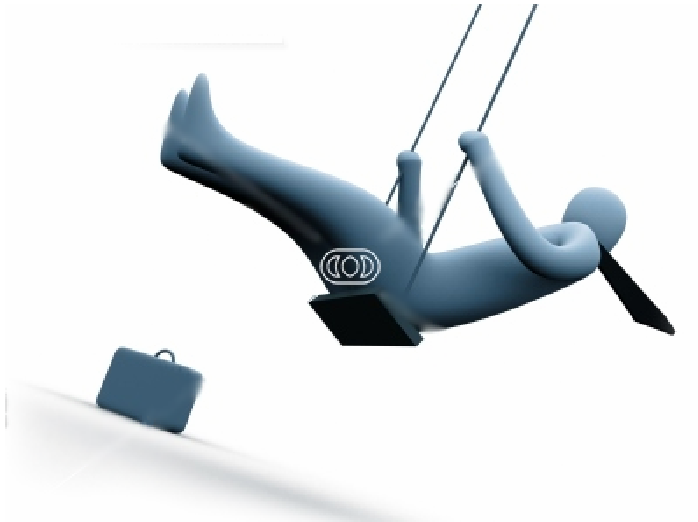
Multiple Negotiations Allowed

Conclusions

Was it all worth it?

- Reduction in traffic associated with negotiation of Resource
- Reduction in user-service interaction
- Extended Agreement gives more power to resource allocation, scheduling, management, aggregation of services

Rizos Sakellariou and Viktor Yarmolenko,
On the Flexibility of WS-Agreement for Job Submission,
Proceedings of the 3rd International Workshop on
Middleware for Grid Computing
MGC '05, vol. 117, 1-6 (Nov. 2005)



http://www.gridscheduling.org