3rd International Workshop on Middleware for Grid Computing
28-29 November 2005
Grenoble
France

On the Flexibility of WS-Agreement for Job Submission

***Viktor Yarmolenko**
Rizos Sakellariou

School of Computer Science
The University of Manchester
Manchester, UK

*corresponding author:
Viktor.Yarmolenmko@manchester.ac.uk

Contents

Introduction & What is Coming

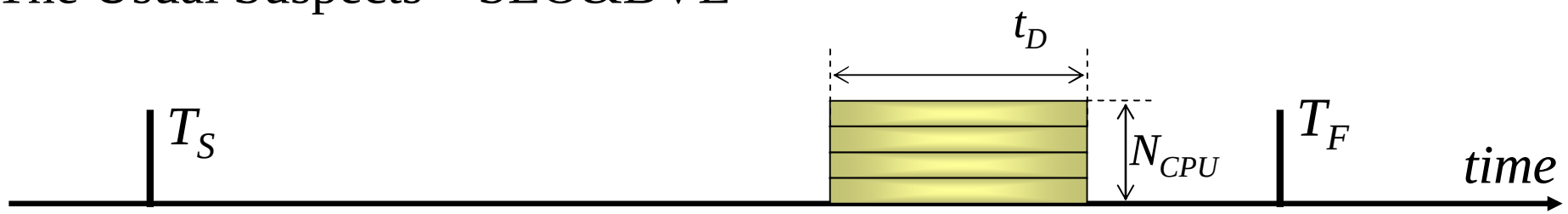
- WS-A Terms: Service Level Objectives & Business Value List
- What are the usual terms for job submission?
- Why WS-Agreement needs extending?
- How do we wan it to be extended?
- Simple scenarios to demonstrate extended WS-Agreement at work
- Simulation model used to prove the point
- What do the results say?

Service Level Agreement (SLA) is nothing more than a contract between two or more parties

WS-Agreement is one of the implementations of SLA

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)*
SLO: B_{job} – *projected traffic that Job creates*

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: $B_{RES}(t)$ – *Resource bandwidth: nominal or @ time*

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

UT: t_S – *actual Job execution start time*

UT: t_{DA} – *actual Job duration time*

UT: $B_{JA}(t_S , t_{DA})$ – *actual bandwidth used by the Job*

UT: $d(n)= n+(n-1)+\ldots+2+1$ – *triangular numbers*

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

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

Guarantee Terms as Functions

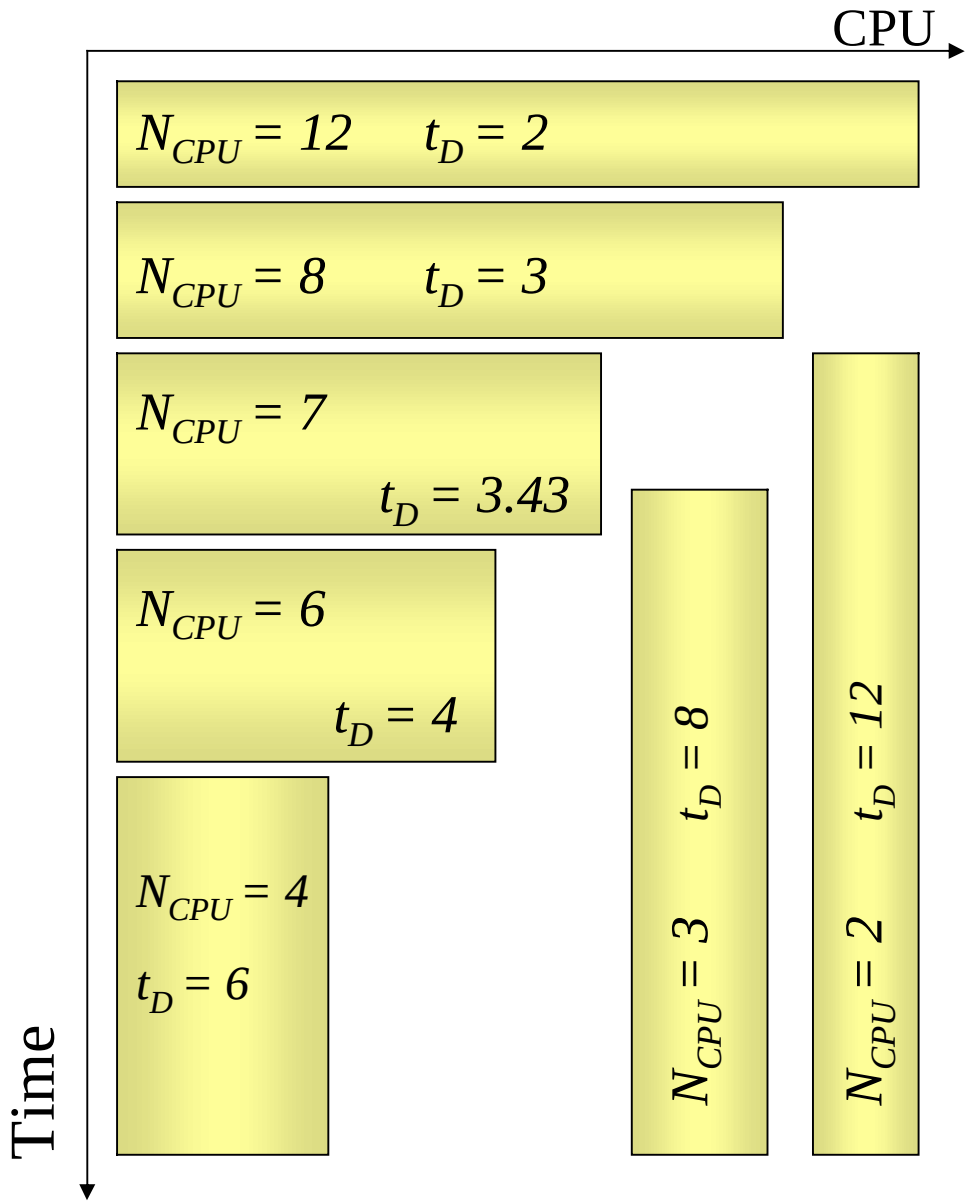
Variable Number of CPUs per Job

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

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

SLO: $t_{UP} = 24$

SLO: $X_{other} = const$



Guarantee Terms as Functions

Adding Variable Bandwidth and Traffic

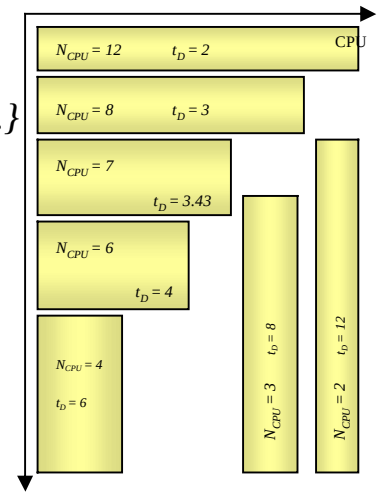
UT: t_{curr}

UT: $B_{RES}(t_{curr})$

UT: $d(n)= n+(n-1)+\ldots+2+1$

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

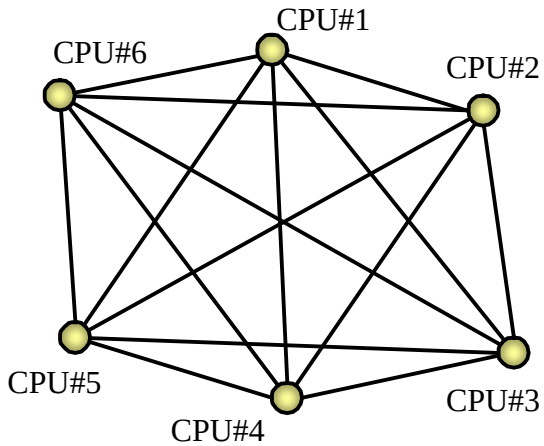
SLO: $t_{UNIPROC} = 24$



SLO: $B_{job} = B_0\, d(N_{CPU} - 1)$

SLO: $t_D = \dfrac{B_{job}\, t_{UP}}{B_{RES}\, N_{CPU}} = \dfrac{B_0\, t_{UP}\ (N_{CPU} - 1)}{2B_{RES}}$

SLO: $X_{other} = const$



For All-to-All topology

Guarantee Terms as Functions

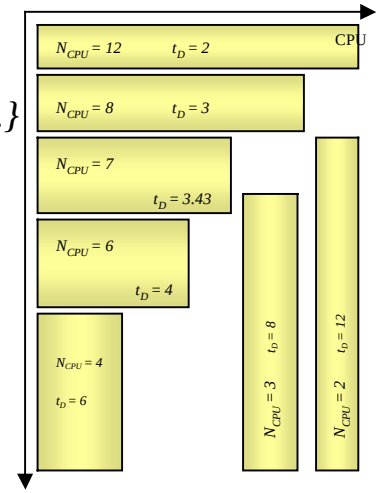
Adding Variable Bandwidth and Traffic

UT: t_{curr}

UT: $B_{RES}(t_{curr})$

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

SLO: $t_{UNIPROC} = 24$

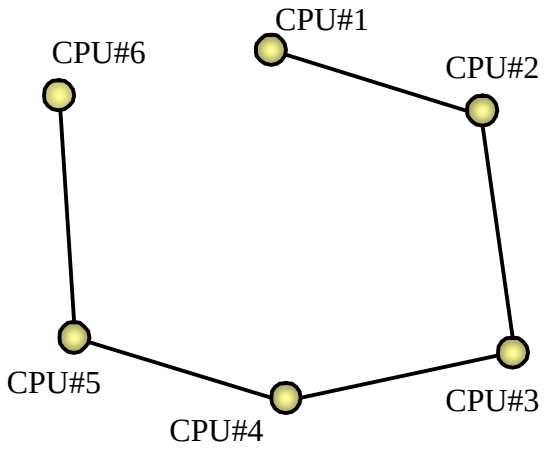


SLO: $B_{job} = B_0 (N_{CPU} - 1)$

SLO: $t_D = \dfrac{B_{job}\, t_{UP}}{B_{RES}\, N_{CPU}} = \dfrac{B_0\, t_{UP}\, (N_{CPU} - 1)}{N_{CPU}\ \ B_{RES}}$

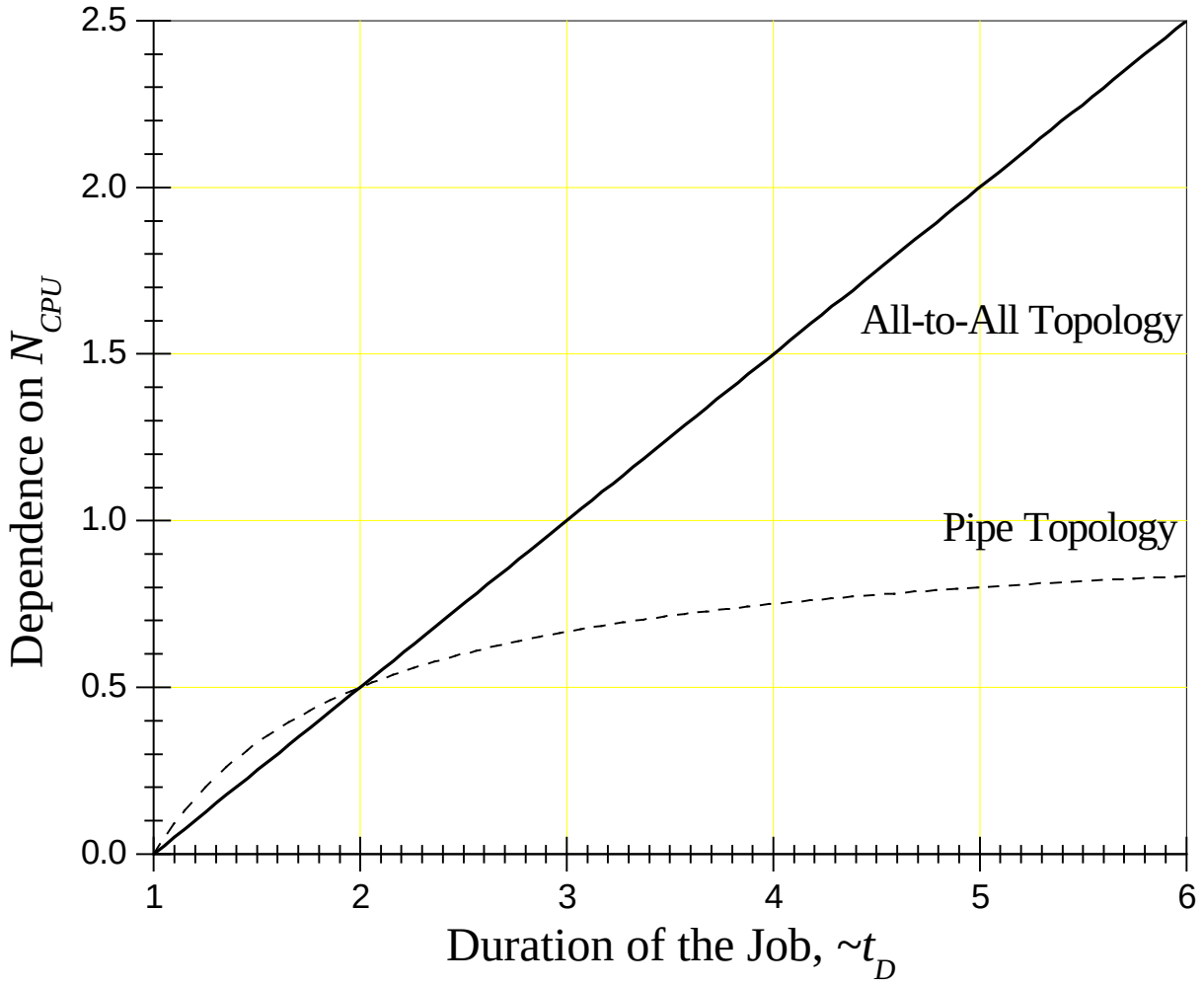
SLO: $X_{other} = const$

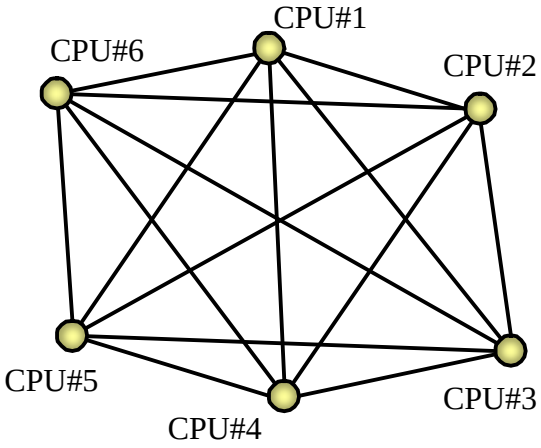
For Pipe topology

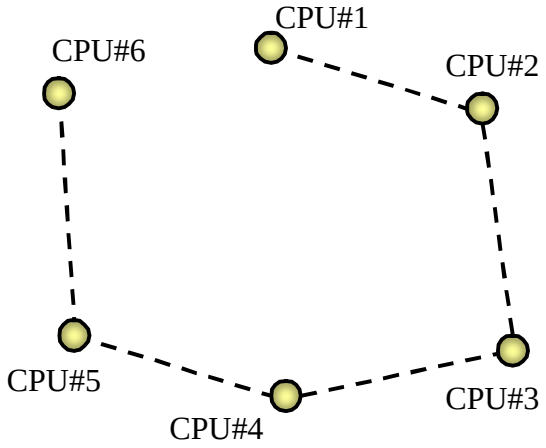


Guarantee Terms as Functions

Comparing the Impact of Two Topologies







Guarantee Terms as Functions

Defining the Value of the Service

UT: t_{curr}

UT: $B_{RES}(t_{curr})$

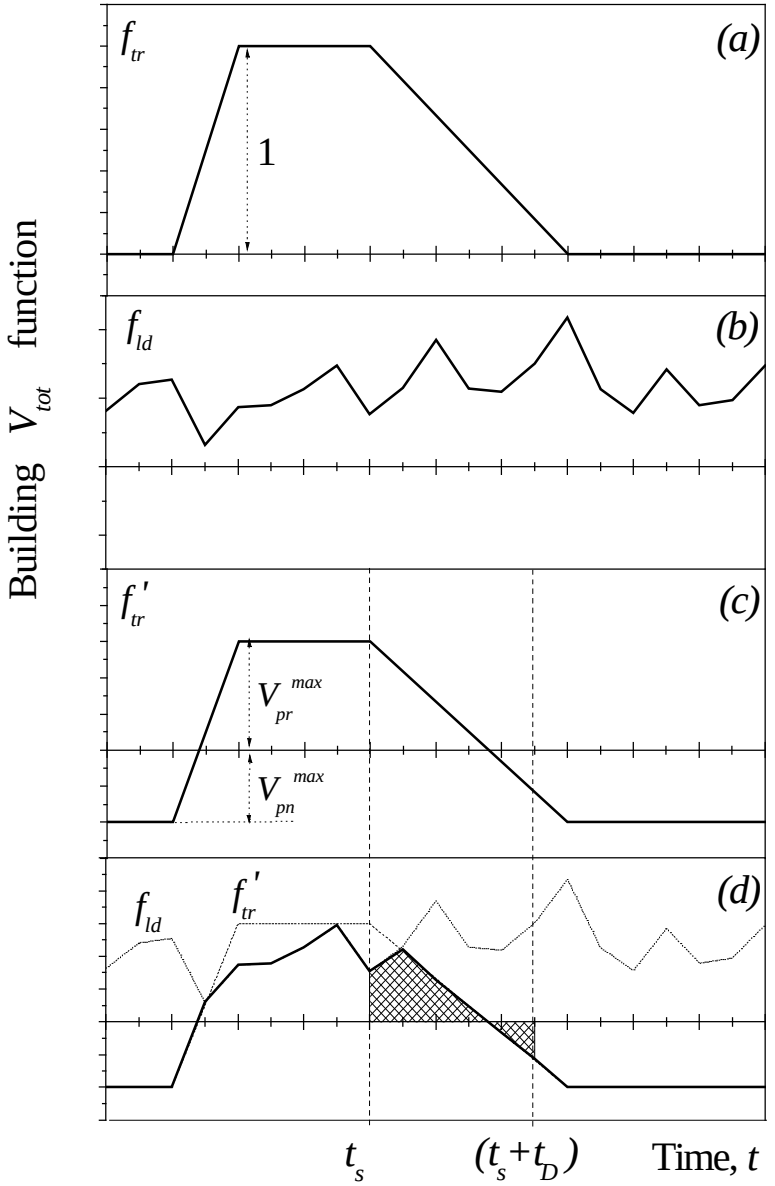
UT: $R_{ld}(t_{curr}) = f_{ld}$

SLO: B_{job}

SLO: $t_D = \dfrac{B_{job}\, t_{UP}}{B_{RES}\, N_{CPU}}$

SLO: $X_{other} = const$

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

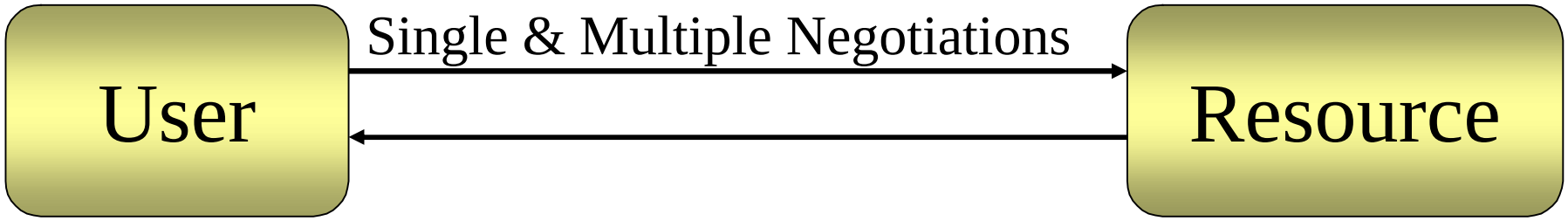


Guarantee Terms as Functions

Suddenly life becomes more interesting

Experiment

The Model



Set of 340 Job requests, for which a solution exists where the 100% utilisation is possible on Resource (147 hours x 64 CPUs)

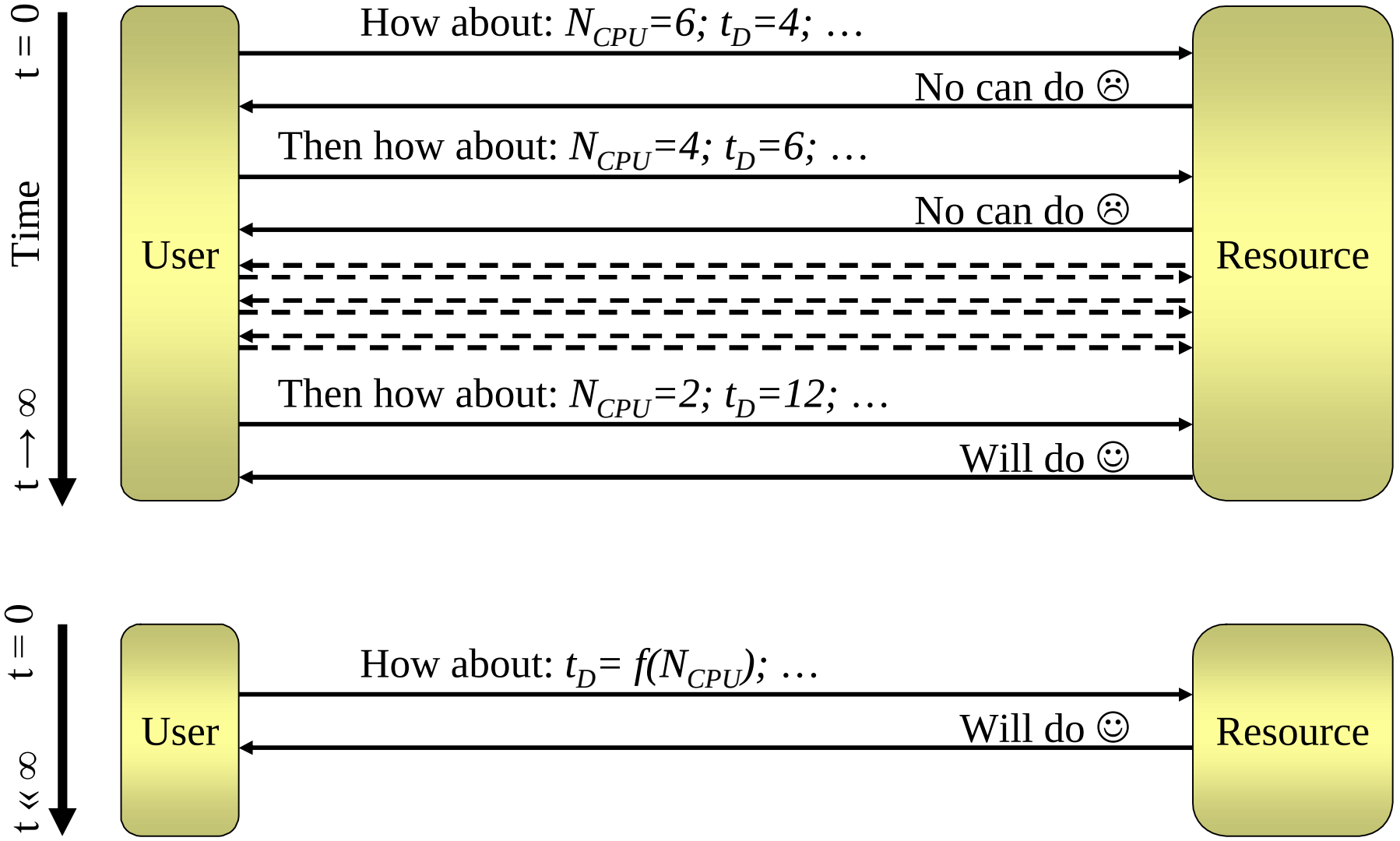
$t_D \times N_{CPU} = A; \ \langle A \rangle = 21.85$

Capacity of 64 CPUs and available for 147 hours

Scheduling by the earliest deadline first (single iteration)

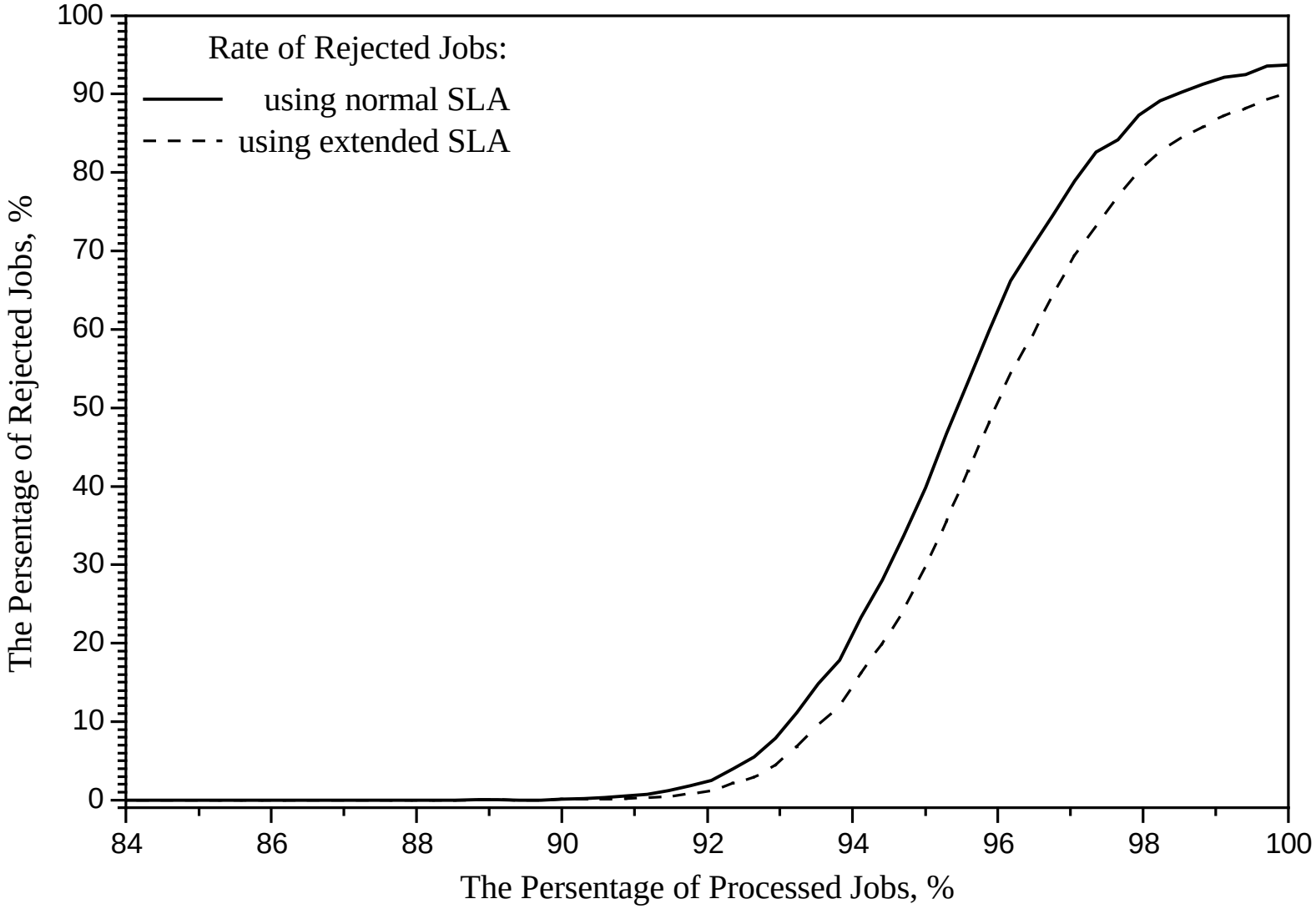
Experiment

Variable CPU Scenario (Original vs. Extended SLA)



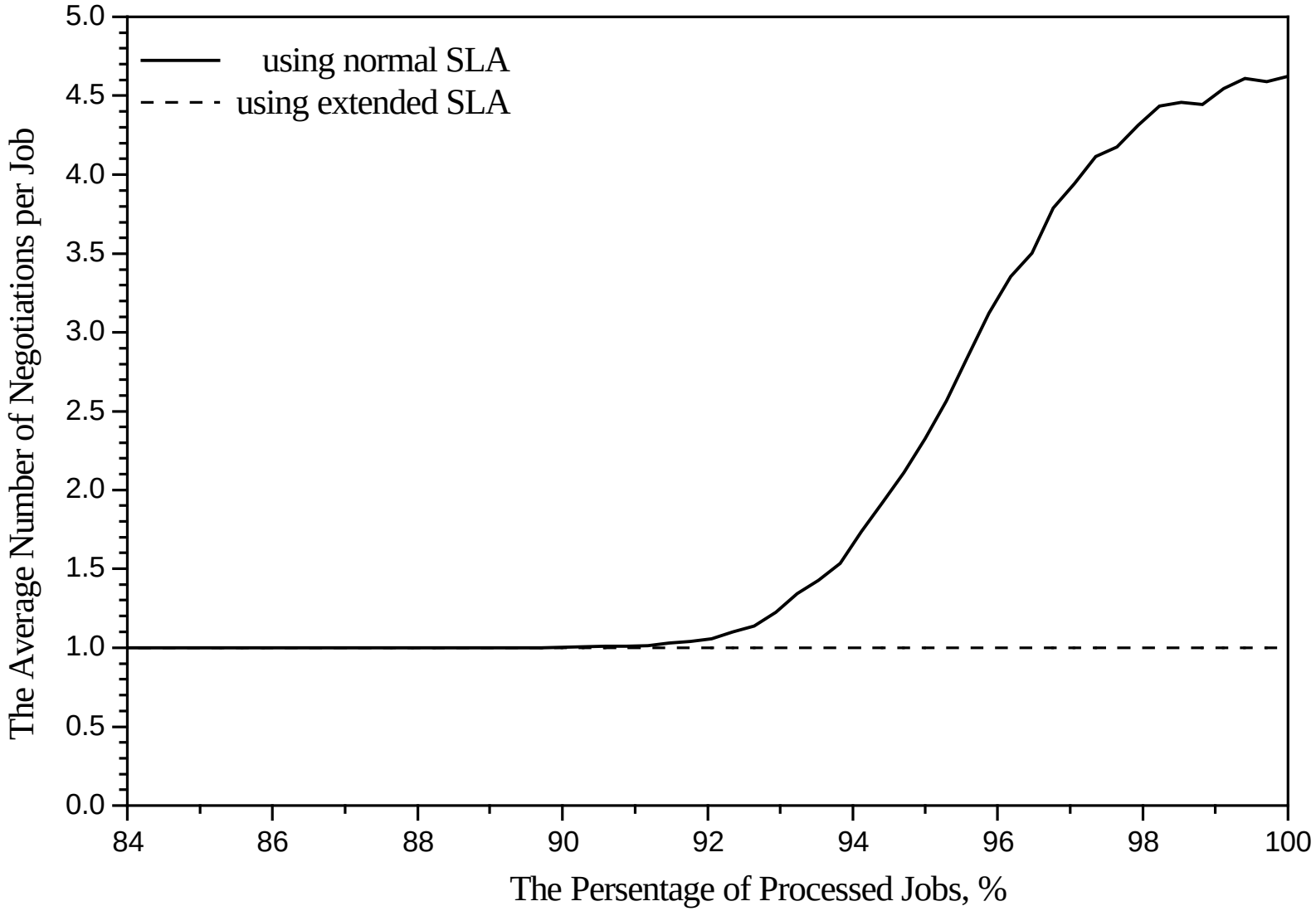
Experiment

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
- Extended Agreement is extensible and could support future demands, e.g. new optimisation algorithms, value added services, autonomous services, …