Negotiation to Execute Continuous Long-Term Tasks

Valeriia Haberland and **Simon Miles** and **Michael Luck**[1]

Abstract. Recently, research has focused on processing tasks that require continuous execution to produce data in a real-time manner. Such tasks often also need to be executed for long periods of time such as years, requiring large amounts of resources (e.g. CPUs) that can be found in a Grid. However, a Grid may be unwilling or unable to allocate resources for continuous usage far in advance, because of high fluctuations in resource availability and/or resource demand. Therefore, a client must relax its requirements in terms of long-term execution, and negotiate a shorter period of execution time; when this period ends, the client must negotiate again to continue task's execution. We propose a negotiation strategy, ConTask, which helps to increase the periods of execution time, and reduce the length of interruptions between them.

1 INTRODUCTION

Much work is being undertaken on processing data streams from sensors, e.g. those that monitor air pollution [5]. It is desirable for these data streams to be processed continuously for long periods in order to control, e.g. environmental parameters such as temperature. The processing of these data streams must avoid interruptions of such length that significant changes to the environment may occur without being noticed. The tasks that process such data streams continuously are considered in our work to be *continuous tasks*. Some research seeks to find an optimal schedule for multiple continuous processes (e.g. continuous mixing of feedstock, packing) to utilise some resources (e.g. equipment) on plants [2], while other research addresses a dynamic area partitioning among a group of robots in order to monitor it continuously in respect of some events of interest (e.g. gas leaks) [1].

A Grid [4] potentially offers more resources (e.g. CPUs) than any supercomputer or cluster. However, a Grid might be unable or unwilling to schedule long-term tasks for the whole requested time duration, because resources could be required by other clients and cannot be monopolised by one client for as long as, e.g. one year. Such task will likely be allocated the execution time which is shorter than it was requested and this task will experience a *planned interruption* when this period ends. A client's role in task execution increases when it can negotiate its requirements with a resource provider, where both negotiators may use different strategies to achieve their goals. In our work, a negotiation is bilateral and starts as soon as a client's task has been interrupted. A *Grid Resource Allocator* (GRA) represents a resource provider and a *Client Agent* represents a client in negotiation, where both are autonomous agents. Time-dependent strategies, e.g. [3,6], naturally express many real-life tasks that depend on time. However, if the end of negotiation is uncertain, but the duration of negotiation is crucial by itself, e.g. for a continuous task's interruption, then the existing time-dependent strategies do not solve this problem.

[1] King's College London, United Kingdom, email: valeriia.kornich, simon.miles, michael.luck@kcl.ac.uk

This paper presents a formal model of continuous task execution and the main ideas behind a negotiation strategy, *ConTask*, which allows a client to relax the last best proposal of the GRA (or its own) to start the next interruption period at the approximate maximum resource availability, where this maximum is estimated based on periodic patterns in resource availability fluctuations. ConTask also considers whether the current duration of interruption, as well as its cumulative duration (including all previous interruptions), is too long. In general, a client becomes more generous in negotiation when an interruption is too long.

2 FORMAL MODEL

This section provides a formal model for continuous execution of a task, including a client's utility. The start time of task execution t_{tot}^{str} is when a client has submitted its first request for resources to a Grid, while the end time for the whole long-term duration of task execution is t_{tot}^{end}. Each task i may have several interruption(s) within $\tau^{tot} = t_{tot}^{end} - t_{tot}^{str}$, which are not simultaneous for multiple tasks. Each adjoined pair $\left(\tau_i^{int}, \tau_{i,l}^{exe}\right)_l$ of a single interruption τ_i^{int} and execution $\tau_{i,l}^{exe}$ periods of time for continuous task i, where an interruption period of time precedes a corresponding execution period of time, has its counter $l = 1, 2, 3, ...$ within τ^{tot}. Here, each $\tau_{i,l}^{exe}$ starts at $t_{i,l}^{str}$ and ends at $t_{i,l}^{end}$. A cumulative duration of interruption $\tau_{i,l}^{cum} = \sum_{k=1}^{l} \tau_{i,k}^{int}$ is the sum of all interruption durations from the first until the lth prior to the execution period $\tau_{i,l}^{exe}$, and it reflects success of the client's strategy over total execution time.

The utility gained by a client for each execution period $\tau_{i,l}^{exe}$ is negatively affected by two factors: a duration of preceding interruption $\tau_{i,l}^{int}$; a cumulative duration of preceding interruptions $\tau_{i,l}^{cum}$. *Effectiveness function* $E_{i,l}(t)$ shows the success level of execution of task i as a process over time within each execution period $\tau_{i,l}^{exe}$, assigning a value from the range $[0, 1]$ to each point in time t. The effectiveness function increases linearly when the task is running (during $\tau_{i,l}^{exe}$) and does not change when the task is interrupted (during $\tau_{i,l}^{int}$). The length of interruptions affects the values of the effectiveness function during the following execution period.

The impact of the length of interruption on the following execution period is calculated with damping functions $SI\left(\tau_{i,l}^{int}\right)$ and $CI\left(\tau_{i,l}^{cum}\right)$ in the range $[0, 1]$ for a single and cumulative interruptions respectively. The values of these functions decrease during the interruption period towards zero. Consequently, these functions are presented as $SI\left(\tau_{i,l}^{int}\right) = 1/\left(e^{\left(\tau_{i,l}^{int} - \tau_{int}^{max}\right)/\epsilon_{int}} + 1\right)$ and $CI\left(\tau_{i,l}^{cum}\right) = 1/\left(e^{\left(\tau_{i,l}^{cum} - \tau_{cum}^{max}\right)/\epsilon_{cum}} + 1\right)$, where τ_{int}^{max} and τ_{cum}^{max} are the durations of single and cumulative interruptions after which an increment of client utility in respect of the following execution period decreases in more than a half of its possible value, compared to the case when an execution period was allocated at once

after a task had been interrupted, while ϵ_{int} and ϵ_{cum} determine the speed of decrease of these functions around τ_{int}^{max} and τ_{cum}^{max}.

The effectiveness function $E_{i,l}(t)$ increases in the range from the level of effectiveness $E_{i,l-1}\left(t_{i,l-1}^{end}\right)$, achieved by a task before interruption $\tau_{i,l}^{int}$ at time $t_{i,l-1}^{end}$, multiplied by the values of functions $SI\left(\tau_{i,l}^{int}\right)$ and $CI\left(\tau_{i,l}^{cum}\right)$, towards the largest possible effectiveness 1 at time t_{tot}^{end} in the time interval $t \in \left[t_{i,l}^{str}, t_{i,l}^{end}\right]$, as below if $j \equiv l-1$:

$$
E_{i,l}(t) = \begin{cases} \left[\dfrac{\left(1 - E_{i,j}\left(t_{i,j}^{end}\right)\right)t + E_{i,j}\left(t_{i,j}^{end}\right)t_{tot}^{end} - t_{i,l}^{str}}{t_{tot}^{end} - t_{i,l}^{str}}\right] \times \\ \times SI\left(\tau_{i,l}^{int}\right) \times CI\left(\tau_{i,l}^{cum}\right) \quad , if\ t \in \left[t_{i,l}^{str}, t_{i,l}^{end}\right], \\ E_{i,j}\left(t_{i,j}^{end}\right) \qquad\qquad\quad , if\ t \in \left]t_{i,j}^{end}, t_{i,l}^{str}\right[. \end{cases}
$$
(1)

Client utility U_i for a task i is the sum of all squares under the effectiveness function for the execution periods in proportion to the maximum possible square when a task does not have any interruptions, and is shown in Equation (2).

$$
U_i = \frac{1}{\tau^{tot}} \sum_{l=1}^{L_i} \left(E_{i,l}\left(t_{i,l}^{str}\right) + E_{i,l}\left(t_{i,l}^{end}\right)\right) \times \tau_{i,l}^{exe},
$$
(2)

where L_i is the total number of interruption-execution pairs within τ^{tot}, and the average utility $U_{aver} = 1/N \sum_{i=1}^{N} U_i$ for N tasks.

3 NEGOTIATION STRATEGY

An objective of our ConTask strategy for a client is to decrease the length of interruptions and increase the length of executions, while not losing significant utility. A client and GRA adopt an *alternating proposals protocol* [8], where the negotiators exchange proposals in turns, and both can either accept the opponent's proposal or quit. In our case, the final deadline of negotiation is unknown, because a client must negotiate with the GRA until it obtains resources, but is time-dependent in terms of the length of interruption. Hence, we use time-dependent strategies [7] with nominally chosen deadlines, and negotiations repeat until an agreement is reached. Traditionally, an agreement is considered to be reached when the negotiator's utility for the current opponent's proposal is higher than for the next possible negotiator's proposal. We propose to take a different approach when a client does not always accept a proposal which is more beneficial than its own next proposal, but may concede further by offering a shorter execution period to the GRA in order to end it at the approximate maximum resource availability, according to its periodic pattern. This decision reflects the continuity of task execution, because a client tends to reduce all possible interruptions within τ^{tot}. The *best offered execution period* $\tau_{i,l}^{exe'}$ is the last proposal of the GRA (or a client), which is to be accepted by a client or which is already accepted by the GRA, but has not yet been confirmed by a client. A client may offer a shorter $\tau_{i,l}^{exe} < \tau_{i,l}^{exe'}$, but only on as long as to reach the closest maximum from the end of $\tau_{i,l}^{exe'}$.

A client may also need to control the durations of interruptions if it is difficult to estimate the maximum resource availability. A client has to decide when single or cumulative interruptions are too long to reach an agreement faster. The values of damping functions $SI\left(\tau_{i,l}^{int}(t)\right)$ and $CI\left(\tau_{i,l}^{cum}(t)\right)$ can be calculated during task's interruption at time t to ensure that their values have not significantly fallen towards 0. Assume that a client defines the *sensitivity thresholds* χ^{int} and χ^{cum} with respect to a single and cumulative interruptions respectively. As soon as the difference between the value of any

damping function and its largest possible value 1 at time t becomes larger than χ^{int} for $SI\left(\tau_{i,l}^{int}(t)\right)$ and / or χ^{cum} for $CI\left(\tau_{i,l}^{cum}(t)\right)$, a client concludes that the interruption is long enough to significantly affect its utility.

A client becomes more or less generous in negotiation, based on our *evaluation function* with three criteria: $C_{i,l}^0(t)$ [7], which shows the risk of resource exhaustion; $C_{i,l}^1(t)$ and $C_{i,l}^2(t)$ (new criteria), which show whether the length of the current single or cumulative interruption is too long. Our evaluation function points out the longest negotiation (a client least generous) among all fuzzy sets combinations [7] when $C_{i,l}^0(t) \geq 0$, $C_{i,l}^1(t) \geq 0$ and $C_{i,l}^2(t) \geq 0$, which means no too long interruptions and no risk of resource exhaustion at time t, or it points out the shortest negotiation (a client most generous) when at least one of the criteria is negative. Table 1 shows the average client utilities over 100 tasks, when the GRA's reservation value (i.e. the longest duration of time it may offer) fluctuates more randomly or more periodically, according to the resource availability changes. The larger its maximal possible deviation, the more random its fluctuations over time. The results show that our ConTask strat-

Table 1. The average client utility for different negotiation strategies.

Maximal deviation,%	ConTask,$\times 10^{-4}$	No ConTask,$\times 10^{-4}$
1	9.93	4.46
10	6.80	3.49
50	4.25	1.67

egy "ConTask" with the sensitivity thresholds 0.2 outperforms our previous strategy "No ConTask" [7] for all deviations.

4 CONCLUSION

This paper described our model of continuous long-term tasks execution in a Grid and, in general, our new ConTask negotiation strategy that allows near-continuity to be maintained among these tasks when they are running by increasing the duration of continuous running and by reducing planned interruptions.

REFERENCES

[1] M. Ahmadi and P. Stone, 'A multi-robot system for continuous area sweeping tasks', in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1724–1729, (2006).

[2] P. M. Castro, A. P. Barbosa-Póvoa, H. A. Matos, and A. Q. Novais, 'Simple continuous-time formulation for short-term scheduling of batch and continuous processes', *Industrial & Engineering Chemistry Research*, **43**(1), 105–118, (2004).

[3] P. Faratin, C. Sierra, and N. R. Jennings, 'Negotiation decision functions for autonomous agents', *Robotics and Autonomous Systems*, **24**(3-4), 159 –182, (1998).

[4] I. Foster, C. Kesselman, and S. Tuecke, 'The anatomy of the grid: Enabling scalable virtual organizations', *International Journal of High Performance Computing Applications*, **15**, 200–222, (2001).

[5] M. Ghanem, Y. Guo, J. Hassard, M. Osmond, and M. Richards, 'Sensor grids for air pollution monitoring', in *Proceeding of the 3rd UK e-Science All Hands Meeting*, (2004).

[6] V. Haberland, S. Miles, and M. Luck, 'Adaptive negotiation for resource intensive tasks in grids', in *Proceedings of the 6th Starting AI Researchers' Symposium*, pp. 125–136, (2012).

[7] V. Haberland, S. Miles, and M. Luck, 'Using adjustable fuzzy inference for adaptive grid resource negotiation', in *Proceedings of the 6th AA-MAS International Workshop on Agent-based Complex Automated Negotiations*, Minnesota, USA, (2013).

[8] A. Rubinstein, 'Perfect equilibrium in a bargaining model', *Econometrica*, **50**(1), 97–109, (1982).