A Token-Based Scheme for Coordinating Decisions in Large-Scale Autonomic Systems

Olga Melekhova, Jacques Malenfant

A Token-Based Scheme for Coordinating Decisions in Large-Scale Autonomic Systems

Olga Melekhova
UC Labs R&D — UC Group
12/14 Rond-Point des Champs-Élysés, 75008 Paris
Email: olga.m@labs.ucgroup.fr

Jacques Malenfant
Sorbonne Universités, UPMC Univ Paris 06, CNRS, UMR 7606 LIP6
4 place Jussieu, F-75005, Paris, France
Email: Jacques.Malenfant@lip6.fr

***Abstract*—Coordinating decisions in large-scale systems is acknowledged as one of the most challenging aspects of autonomic computing. If control theory is already pushed to its limits by the inevitable decentralised nature of such systems, going large-scale makes things even harder. In this paper, we propose a decentralised token-based scheme for coordinating shared resources allocation in large-scale autonomic systems. In this scheme, each token represents an exclusive access to a share of the resource. Tokens are used and exchanged among autonomic managers to limit their decisions to the resource capacity the tokens they hold at this moment represent. Our token-based protocol is meant to provide statistical guarantees on the average total resource usage and the average lateness of node actions due to the coordination. It exposes a set of parameters which we have explored to characterise its fairness and global resource usage on a use case in geotracking. Experiments with the coordination of 10.000 autonomic managers have shown very good results for large spectrum of parameter values and system's regimes.**

I. Introduction

The enormous growth in complexity of computer systems has pushed IBM to propose a comprehensive effort, called *autonomic computing*, to automate their management [1]. In IBM's vision, autonomic managers implement the well-known MAPE-k control loop from its architectural blueprint [1]. Issues relevant to the implementation of isolated autonomic managers are now well-studied, but as computer systems are more and more decentralised, coordination is needed between autonomic managers that adapt dependent managed elements [2]–[4]. To sustain large-scale, new schemes are needed that allow a large number of autonomic managers to coordinate effectively with each others. Kephart pointed out that coordination in large-scale autonomic systems is one of the hardest challenges in this field [5].

Known patterns in distributed decision-making do not scale as they would require an ever longer diffusion of information over an ever larger set of autonomic managers. Contrary to other contexts, time is a crucial issue in autonomic computing, as decisions are part of feedback control loops that must strictly follow the dynamics of the system. In control theory, the stability and the controllability properties crucially depend on timely decisions [6]. Making available the necessary information in bounded time to all of the autonomic managers is therefore crucial to the overall behavior of the system.

In this paper, we address the problem of coordinating decisions that are system-wide implicitly dependent, the consumption of a central shared resource. The objective is to make sure that autonomic managers take decisions at the pace of changes in the need of their managed elements while the total amount of required resource never exceeds its total capacity. The coordination must provide for statistical guarantees on the resource usage and on its fair distribution during the whole execution of the system. Coordination can be made to *emerge* [7]–[9] from local decisions, but broadcasting the current levels of resource usage of all managed elements to every autonomic manager would inevitably breach the time constraints of local control loops as the system scales.

We rather propose a token-based scheme, inspired from decentralised mutual exclusion algorithms [10]. The total amount of available resource is divided up in tokens that represent an exclusive right to use the corresponding amount of resource. Autonomic managers are organized into an overlay network [11] over which tokens circulate using random walks [10]. When an autonomic manager takes a decision that implies a raise in resource consumption, it first limits it to the amount represented by tokens it holds and then try to acquire new tokens to raise it further. When the autonomic manager takes a decision that lowers the consumption, tokens are reinjected in the overlay network for other autonomic managers.

Fair distribution of the resource among autonomic managers and the best possible global usage of the shared resource are properties highly influenced both by the density of the overlay network and by the fluidity in the flow of tokens. The first issue was addressed in [12]. In this paper, we explore in depth the second issue, and we propose a scheme exposing relevant parameters and show how to set their values to balance the above objectives and to scale. We report on experiments on systems of 10.000 autonomic managers that validate the approach and provide insights to tune the parameters to get the best possible coordination.

The rest of the paper is organized as follows. Section II covers the design of our scheme and the related work. Section III digs into the token-based scheme itself, while Section IV presents the experimental results. A conclusion ends the paper.

II. Coordinating decisions in autonomic systems

A. Coordination

Large-scale coordination of autonomic managers can be seen as a dynamic distributed decision-making problem. Centralised solutions obviously do not scale. To scale, decisions must be made locally, and then coordinated by disseminating the information to every decision-making sites. But collecting locally all of the information at decision-time does not scale when the pace of decision-making is faster than the time required for the collection. Asynchronous collection of information from the decision-making do not scale either when the time for information to travel from its generating site to the one needing it exceeds the deadlines of local decisions it impacts. In autonomic computing, where decisions must obey strict deadlines, the decoupling must ensure the local availablility and freshness of data within strict time constraints.

In order to scale, autonomic manager coordination must rely on the smallest amount and most locally possible exchanged information. Token-based coordination (e.g. [10]) takes an interesting stance at the information dissemination problem by reversing viewpoints: in a coordination for resource sharing, instead of disseminating the resource usage from all the nodes to every others, a token represents a share of the resource than no other peers can be currently using. Hence, the capture of a token ensures the exclusive access to that share of the resource and for as long as the peer keeps it.

To get good resource usage and fairness, autonomic managers must be forced to release tokens after use and released ones be acquired as soon as possible by autonomic managers that need them. The latter issue is addressed by organizing autonomic managers in an *overlay network* [11] specifically designed for each kind of target systems, allowing to efficiently move released tokens to the autonomic managers that need them. To force token release, the system must impose rules for capturing, keeping and releasing tokens to every autonomic managers. In real systems, indeed, careful monitoring and penalties should discourage autonomic managers not doing so.

To sum up, the general token-based approach to large-scale coordination decomposes into three steps:

1) Define a suitable semantics of tokens with regards to the local decision-making processes.
2) Link autonomic managers through a specific overlay network and adopt a suitable token communication policy.
3) Make the local decision-making able to execute within its deadlines, on the basis of tokens received prior to each decision-making instant.

In the rest of the paper, we apply this idea to a use case in geotracking systems. The issues and our solutions to construct an overlay network appropriate for this use case have been proposed ealier [12]. Section III covers the issues revolving around the design of a token-based protocol providing the necessary information to a local decision-making algorithm.

B. A use case in geotracking

Our use case considers the geotracking of large fleets of trucks coming from the French ANR project SALTY [13]. Geotracking is now crucial to several industries, as logistics, to optimise the management of fleets and to cater for security in high-risk transportation (currency, radioactive material, etc.). Typically, geotracking is used to follow trucks as to provide users with alerts as soon as they exit a predefined corridor or, perhaps more commonly, when approaching logistic bases to trigger resource allocation (doors, personal) for their arrival.

In this context, we strive to set in real-time the delays between position sendings for each of the GPS. Making the least possible sendings while achieving the geotracking objectives is important to lower the costs for users. But another objective comes from the specifics of the project. Our industrial partner offers value-added services by making its customers receive their alerts through a unique centralised position processor called GeoHub (the term belongs to Deveryware). We have leveraged the issue of maintaining the quality of service of this position processor at constant level, by making it a large-scale coordination problem defined as follows: *How to set the delays between position sendings for each truck's GPS in such a way that the overall frequency of position receptions on the processor never exceeds its maximum processing capacity?*

Operators of fleets have trucks with GPS sending positions to GeoHub through GSM. They are followed by geotracking applications, seen as managed elements, receiving alerts from the GeoHub upon position receptions and evaluations. We put over these applications autonomic managers that also receives positions from the GeoHub and state information from applications to choose the delay to the next position sending; such orders also go through GeoHub to the GPS. When no imminent alert is expected, longer delays are chosen, but when trucks approach a trigger, shorter ones are chosen to catch the potential alert within the required precision.

To enforce the GeoHub quality of service, autonomic managers use tokens having a fixed period between successive usages representing a share of the overall GeoHub processing capacity. When deciding for a delay to their next position sending, autonomic managers cap it to the earliest next time of usage of a token they hold at decision-time, but they can revise this decision if a better suited token can be acquired before the sending is done (see Section III for details).

C. Related work

Coordination in large-scale distributed systems has attracted much attention during the last decades. Today, perhaps the most promising approaches are bio-inspired algorithms where global behavior *emerges* from purely local decisions [7]; some applications have been conceptually explored [9] and only recently experimented for the first time in autonomic computing [14]. Crucially depending on information flows [15], emergence approaches for large-scale autonomic computing are also limited by the convergence delays in the broadcasting of information and in reaching an overall stability [16], [17]. Indeed, these approaches still require a lot of experiments, to which our work claims to contribute.

Louleiro *et al.* [18] address a large-scale resource allocation problem closely related to ours. In their setting, entities need shared resources to process their workload, and their relative importance is modeled by utility functions of all workloads. Autonomic managers gather each others workloads, and when the workload of an entity raises, its autonomic manager uses its utility function to find the right decision. The authors use an *epidemic protocol* [16] to broadcast the individual workloads to all entities. Simulated on overlay networks of 500 entities, with degrees between 5 and 7, their approach requires more than 10 cycles of message exchanges between all of the nodes to propagate the information over the whole network. The number of cycles can be diminished by augmenting the degrees, but the number of messages per cycle would also augment. Picard *et al.* [19] propose a similar solution for distributed constraint satisfaction, without experimenting a dynamic version of the problem though.

Other works on protocols for resource discovery in grids have shown that broadcasting approaches have an exponential complexity in the number of messages [20], [21]. Clearly, obeying strict timing constraints when scaling requires a stronger decoupling between local decision-making and information dissemination. Our contribution is therefore to study and experiment with new approaches towards this goal.

Few related works address decision-making in the context of large-scale decentralised control where the system has strict timing constraints. Decentralised, networked control is still an open problem in control theory [22], even at the small-scale. Nevertheless, efforts to coordinate decisions in distributed autonomic systems have been made. IBM's teams have tackled the problem of energy management in data centers, but for small-scale systems only [2]. From these experiments, Kephart [5] argued recently that scaling is still a central challenge. Work from Oliveira *et al.* [4] on synchronising autonomic control loops also targets the small-scale and does not take into account the timing issues of control. Many of the other works published recently are in the same vein [3], [23], [24].

Coordination in general has been studied in many fields, among which multi-agents systems (MAS), to such an extent that it is impossible to cover all of them. Compared to our proposal, most of these works address relatively small-scale systems or do not consider the timing issues of control [25]. For example, Gerber and Jung [26] consider a solution where local decisions are taken without delays, but use a global reasoning coordinating them that would hardly scale. Rustogi and Singh [27], as us, use local decision-making and show that giving locally more global information does not always lead to better decisions, concurring to more decoupling between local decision-making and information dissemination.

In the field of decision-making, reinforcement learning for multiple decision-makers has also been studied as coordination mechanism, but only for a few decision-makers [28]. More recently, *collaborative reinforcement learning* (CRL) [29]–[31] over larger sets of decision-makers has been proposed, but scaling still requires experiments, as the current works still consider a few tens of them only. However, CRL remains a very interesting approach, as it promises to completely decouple coordination acts from coordination policy computing, hence making possible zero-delay local decisions.

To summarize, our contribution is the design and the implementation of new approaches and techniques for decentralized decision-making subject to time constraints, more or less strict.

III. Token-based coordination

A. General model and issues

Based on our overlay network algorithms introduced in [12], nodes exchange tokens controlling the (quantitative) access to the central shared resource, the GeoHub. As the decisions of the different autonomic managers do not depend upon any other managers but upon the whole, the overlay network can be random. As individual autonomic managers do not know to which autonomic manager its released tokens should be sent, tokens circulate on the overlay network by random walks *i.e.*, a token message does not identify its destination but is rather sent to a randomly selected neighbour which either captures it for its own usage or immediately releases it and sends it again to a randomly selected neighbour among its own ones. Each token corresponds to the same rate of position sendings. To enforce its rate, the token has a period p_j and an earliest time of next utilisation t_u; each time $t \geq t_u$ a token is used to enable a position sending, its t_u is reset to $t + p_j$.

Token-based algorithms have been used to solve a distributed version of the k-mutual exclusion problem [10] *i.e.*, at most k processors can enter their critical section at any time. Tokens circulate using random walks over a network of processes, and the processes must capture a token to enter their critical section. Our scheme and some of our solutions are inspired from these, though the scale of the considered systems are different. Also, tokens in k-mutual exclusion schemes represent a binary access right with $k \ll N$ in general, while in ours they represent a quantity of resource. Hence our scheme typically use much more tokens and acquiring another one after a capture makes sense, *e.g.* if it allows to make the sending earlier, contrary to mutual exclusion. Therefore, properties and parameters of the algorithms are quite different and as such, need new experiments.

The lack of space in this short paper forces us not to address issues related to failures in the network, but we did tackle crash failures and to some extent transient ones.

B. Coordination algorithms

The coordination decomposes in two main coordinated decision-making: handling position receptions and handling token arrivals. The coordinated decision-making occurs first when a position is received from the GPS of the truck through GeoHub, at which time the local decision-making process proposes a delay to the next position sending according to the needs of the geotracking applications and tries to plan it if the necessary token is available. When a token arrive, the autonomic manager must decide if it can capture or reemit it, and in the case of capture, plan a new position sending according to the most recently required time for it. To minimise the waiting time for a token when an autonomic manager needs one and one is available somewhere, we propose to use a unique parameter to control their fluidity, namely the time period τ_j during which a token can be kept before using it. The larger τ_j is, the more likely a token can be kept or captured.

Figure 1 illustrates the different important instants for some illustrative scenario. At time t_d^i, a position is received and the

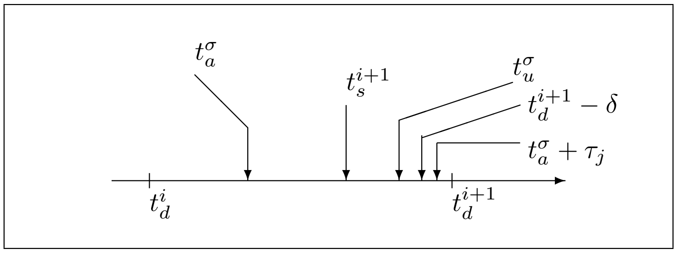

Fig. 1. Main instants when deciding and handling arriving tokens.

τ_j : maximum keep time of a token before usage
f_j : tokens' position sending frequency
p_j : the period of tokens ($p_j = 1/f_j$)
ϵ : minimal delay between a decision and the sending
δ : minimal modification in the time of the next sending justifying its change
t_u^j : earliest time for the next utilisation of token j
t_d^i : time of the i^{th} local decision
t_s^{i+1} : requested time for the next position sending
t_d^{i+1} : decided time for the next position sending

Fig. 2. **Common data** (algorithms of figures 3 and 4)

```
1: $\alpha \leftarrow$ the token captured for this sending
2: $t_u^\alpha \leftarrow t_d^i + p_j$ {mark the token for the current sending}
3: {look to release the token, if required}
4: if $t_u^\alpha \leq t_d^i + \tau_j \wedge t_s^{i+1} \leq t_d^i + \tau_j$ then
5:     keep the token $\alpha$
6:     $t_d^{i+1} \leftarrow max(t_s^{i+1}, t_u^\alpha, t_d^i + \epsilon)$ {next sending}
7: else
8:     release the token $\alpha$
9:     $t_d^{i+1} \leftarrow \infty$ {no token left, so no next sending}
10: end if
```

Fig. 3. Position sending at t_d^i with token α

```
if $t_s^{i+1} \leq t_d^{i+1} - \delta \wedge t_a^\sigma + \epsilon \leq t_d^{i+1} - \delta \wedge t_u^\sigma \leq t_a^\sigma + \tau_j$
        $\wedge\, t_u^\sigma \leq t_d^{i+1} - \delta \wedge t_s^{i+1} \leq t_a^\sigma + \tau_j$ then
    if a token has been captured since $t_d^i$ then
        release the previous token
    end if
    keep the token $\sigma$
    $t_d^{i+1} \leftarrow max(t_s^{i+1}, t_u^\sigma, t_a^\sigma + \epsilon)$
else
    release the token $\sigma$
end if
```

Fig. 4. Token σ arrival at t_a^σ

local decision-making proposes a delay for the next position sending that should occur at t_s^{i+1}. The coordinated decision is then to plan the next sending at $t_d^{i+1} \geq t_s^{i+1}$ that corresponds to the t_u of the token the autonomic manager holds, if any. Figure 1 shows a scenario where the autonomic manager holds a token α with $t_u^\alpha > t_s^{i+1}$, so $t_d^{i+1} \leftarrow t_u^\alpha$ and therefore $t_d^{i+1} > t_s^{i+1}$. Later on, when a token σ arrives at t_a^σ, the autonomic manager captures it only if the conditions over τ_j hold and if it can use it to advance or better match t_d^{i+1}. The presented scenario illustrates the case where $t_a \leq t_s^{i+1} \leq t_a^\sigma + \tau_j$ and $t_u^\sigma \leq t_a^\sigma + \tau_j$, which implies that the token can be used to advance t_d^{i+1}, hence it can be captured. Pragmatically, modifying t_d^{i+1} involves sending an order to the GPS through the GeoHub, hence the change must worth this effort (and respect a minimal delay). This requirement is formulated as the constraints $t_s^{i+1} + \epsilon \leq t_d^{i+1} - \delta$ and $t_u \leq t_d^{i+1} - \delta$ for some predefined ϵ and δ representing the shortest delay for modification and the smallest allowed one.

Figure 3 shows the coordinated decision at position reception time. First (2), the token previously captured to "cover" the current sending is marked as used by resetting its t_u^α to $t_d^i + p_j$. Next, the token is looked up to see if it must be released. It will be kept only if it can be used for the next required sending (4-6), *i.e.* if the token can be used and the next sending planned within the next τ_j period of time, or more formally if $t_u^\alpha \leq t_d^i + \tau_j \wedge t_s^{i+1} \leq t_d^i + \tau_j$. If the token is kept (5), the next position sending can be planned (6). Otherwise, the autonomic manager will have to wait for the arrival of another, capturable token (line 9). Line 6 also shows how the pragmatical parameter ϵ is used to force the next position sending to occur in at least ϵ seconds. Figure 4 shows the coordinated decision at token arrival time. To capture a token, the autonomic manager must need one and/or be able to use it during the next τ_j period. Hence, the following conditions must be obeyed to replan the next sending:

- $t_s^{i+1} \leq t_d^{i+1} - \delta$ *i.e.*, sending previously planned at t_d^{i+1} is allowed to be replanned;
- $t_a^\sigma + \epsilon \leq t_d^{i+1} - \delta$ *i.e.*, replanning is feasible;
- $t_u^\sigma \leq t_a^\sigma + \tau_j$ *i.e.*, the token can be used within τ_j;
- $t_u^\sigma \leq t_d^{i+1} - \delta$ *i.e.*, be used for that replanning;
- $t_s^{i+1} \leq t_a^\sigma + \tau_j$ *i.e.*, next sending can occur before τ_j.

IV. Experimental results

A. Experimental setup

Because controlled field experiments with real trucks was just too complex and costly, our experimental setup uses a discrete-event simulation of the GeoHub to receive requests for position sendings from autonomic managers and push them data at the requested moment. Pushed data is a forecast delay until their next potential geotracking alert *i.e.*, random numbers generated in such a way to control the overall level of resource required and see how well the coordination is doing with it.

Autonomic managers are implemented as concurrent distributed components in Java/RMI. In the runs which results are reported here, 10.000 autonomic managers were distributed among 50 JVM (Oracle Java7u51 on MacOs X 10.9), themselves running on 5 computers (Mac mini, 2.6GHz Intel i7, 16Gb 1600MHz DDR3) connected through Ethernet. Another computer (Mac pro mid-2012, with 2×2.4GHz Intel Xeon, 24Gb 1333MHz DDR3 ECC) runs the GeoHub simulator.

Thanks to our previous overlay network management algorithms, the network topology was very stable from runs to runs, each autonomic managers having an average of 15.5 neighbours, with a standard deviation of 1.5.

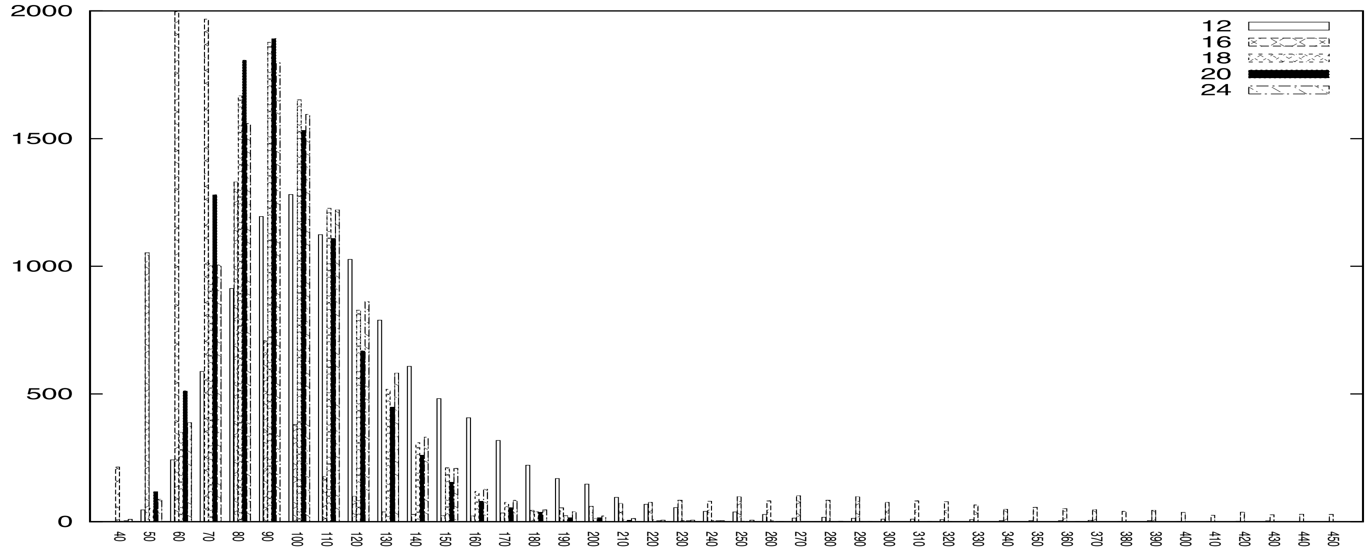

Fig. 5. Mean latenesses distribution for $k = 800$ and $\tau_j = 12, 16, 18, 20, 24$, x goes from 40 to 450 seconds of lateness and y from 0 to 2000 autonomic managers.

B. The core experiments

The core experiments were targeting two major questions: is the token-based coordination effective and fair enough, and can we characterise to some extent the spectrum of parameters that work well? Effectiveness was assessed as the average global resource usage as well as its standard deviation. Fairness was assessed by measuring for each autonomic manager the average lateness in its position sendings (*i.e.* the delay between the requested time and the actual time of sending), and then we have looked at the distribution, the mean and the standard deviation of these average latenesses.

We have evaluated three main system's regimes: overloaded, even and underloaded. In each of them, the capacity of the Geohub has been fixed at 50 positions processed per second (*pps*). In the overloaded case, each autonomic manager requires one position sending with delays being randomly generated using a N(100, 50) distribution. This gives an average global pressure of 100 pps, near twice the capacity. The even scenario uses a N(200, 100) distribution, exhibiting a global pressure of 50 pps, equal to its capacity, while the underloaded uses a N(300, 150), giving a pressure of 33.3 pps.

As the number of tokens can influence the coordination, we have experimented with 400, 800, 1200 and 1600 tokens, giving token periods $p_j = 8, 16, 24$ and 32 seconds. Our previous work [12] has shown through simulations that the coordination works well when $\tau_j \approx p_j$. So experiments used as values of τ_j roughly $0.75p_j$, p_j, $1.125p_j$, $1.25p_j$ and $1.5p_j$. Each run lasted 1.5 hours. To compute the distribution of the rate of positions actually processed, we have measured it by periods of 30 seconds and then computed the average and standard deviation of these values. To compute the mean of lateness means, we have computed the mean for each autonomic manager, and then we have computed the mean of these means and the standard deviation.

Results of the 60 runs show that good results are obtained for values of τ_j between p_j and $1.5p_j$, but slightly dropping as τ_j increases over $1.25p_j$. The influence of the number of tokens appears less important, but a too low or a too large number of tokens does not seem to help, especially for small τ_j. Under the overloaded scenario, the global usage of the resource can be very close to the optimal 50 pps (99.56% for 1600 tokens and $\tau_j = 32$ s. Coarsely speaking, we obtained a resource usage between 95% and 99.5% for $16 \leq \tau_j \leq 48$ with the proper setting of the number of tokens between 800 and 1600 in each case.

The distribution of the mean latenesses also show a very good fairness, though it should be enhanced and finding new criteria for capturing tokens that would decrease the standard deviation and increase the fairness is certainly a goal for future research. Figure 5 shows the distributions for 800 tokens for the different τ_j. The mean lateness ranges from 40 to 450 seconds. For larger τ_j the distributions are very similar and relatively grouped, while for the smaller ones they tend to be less grouped, with the one for $\tau_j \leq 16$ having a somewhat longer tails, hence favouring $\tau_j > p_j$.

The even and underloaded scenarios make it easier to assess the loss of resource imposed by the algorithm. In the even scenario, we can see a larger standard deviation in global resource usage and a larger relative standard deviation in the mean of mean latenesses. This is due to the fact that when the pressure on the central resource is no longer controlled, when under 50, the full randomness of the requested positions shows directly. This is even more the case in the underloaded scenario, where basically no control is imposed by the tokens, but only a lower performance due to the potential unavailability of tokens at the right moment. Hence the means in these two cases are more important than the standard deviations. And we can observe that in the underloaded scenario, the mean resource usage and the mean of mean lateness are close to 33.3 and 0, showing a low impact of the token-based scheme *per se*.

V. Conclusion

In this paper, we have presented a new token-based scheme for large-scale coordination in decision-making for autonomic systems that provides for statistical guarantees on the quality and fairness of the coordination. The coordination addresses the fair distribution of a central resource of limited capacity among autonomic managers adapting managed elements in a way that changes their resource consumption. In our geotracking use case, it allows more precisely to coordinate the decisions of autonomic managers adapting the rate of position sendings for their managed element while making sure that the overall rate of position processing does not exceed the total processing capacity. This capacity is divided up into tokens of fixed period circulating by random walks on an overlay networks of autonomic managers. When autonomic managers need to plan the next position sending for their managed element, they must hold or capture the necessary tokens.

Our main contributions lie in the quantitative nature of tokens experimented in our coordination scheme, in a novel simple local rule to capture and release tokens, and in large-scale experiments on systems of 10.000 autonomic managers. Contrary to token-based distributed k-mutual exclusion where tokens represent a binary access right, our tokens represent available shares of quantitative resources. After experiments showing that fluidity in the circulation of tokens is essential to a good coordination, we have proposed a simple parameter τ_j to control it. τ_j is the delay during which a token can be kept before using it to access the resource. Experiments on our geotracking use case have shown that, on an overloaded scenario, our scheme can achieve as much as 98–99% in the global usage of the shared resource with a standard deviation of less than 1%. Fairness is also very good, the ratio standard deviation over average being kept under 25%. Overall, very good coordination results are obtained for a wide spectrum of parameter values and system's regimes.

Acknowledgments

Work partly funded by the ANR project SALTY (ANR 09 SEGI 012 08).

References

[1] IBM, "An Architectural Blueprint for Autonomic Computing," IBM Corporation, White Paper, April 2003, 1^{st} Edition, 4^{th} in June 2006.

[2] J. Kephart, H. Chan, R. Das, D. Levine, G. Tesauro, F. Rawson, and C. Lefurgy, "Coordinating Multiple Autonomic Managers to Achieve Specified Power-Performance Tradeoffs," in *ICAC'07*, June 2007.

[3] P. Vromant, D. Weyns, S. Malek, and J. Andersson, "On Interacting Control Loops in Self-Adaptive Systems," in *ACM SEAMS'11*, 2011, pp. 202–207.

[4] F. A. de Oliveira Jr., R. Sharrock, and T. Ledoux, "Synchronization of Multiple Autonomic Control Loops: Application to Cloud Computing," in *Coordination'12*, ser. LNCS, vol. 7274. Springer-Verlag, 2012, pp. 29–43.

[5] J. O. Kephart, "Engineering decentralized autonomic computing systems," in *SOAR'10*. New York, NY, USA: ACM, 2010, pp. 1–2.

[6] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback Control of Computing Systems*. Wiley Interscience, 2004.

[7] L. Steels, "Towards a Theory of Emergent Functionality," in *From Animals to Animats (First Int. Conf. on Simulation of Adaptive behaviour)*. MIT Press, 1990, pp. 451–461.

[8] T. D. Wolf and T. Holvoet, "Emergence as a General Architecture for Distributed Autonomic Computing," Department of Computer Science, KU Leuven, Tech. Rep. CW384, 2004.

[9] ——, "Emergence versus self-organisation: different concepts but promising when combined," in *Engineering Self Organising Systems: Methodologies and Applications*, ser. Lecture Notes in Computer Science, vol. 3464. Springer-Verlag, 2005, pp. 1–15.

[10] A. Israeli and M. Jalfon, "Token Management Schemes and Random Walks Yield Self-Stabilizing Mutual Exclusion," in *ACM Symposium on Principles (PODC'90)*, 1990, pp. 119–131.

[11] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, pp. 72–93, 2005.

[12] O. Melekhova, "Coordination dans la décision pour les architectures autonomiques à grande échelle," *Technique et science informatiques*, vol. 33, no. 1-2, pp. 31–62, February 2014.

[13] O. Melekhova, M.-A. Abchir, P. Châtel, J. Malenfant, I. Truck, and A. Pappa, "Self-Adaptation in Geotracking Applications: Challenges, Opportunities and Models," in *ADAPTIVE 2010*. IEEE, 2010, pp. 68–77.

[14] T. D. Wolf, G. Samaey, T. Holvoet, and D. Roose, "Decentralised Autonomic Computing: Analysing Self-Organising Emergent Behaviour using Advanced Numerical Methods," in *IEEE ICAC'05*, 2005.

[15] T. D. Wolf and T. Holvoet, "Designing Self-Organising Emergent Systems based on Information Flows and Feedback-loops," in *IEEE SASO 2007*, 2007.

[16] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-Based Aggregation in Large Dynamic Networks," *ACM Transactions on Computer Systems*, vol. 23, no. 3, pp. 219–252, August 2005.

[17] I. Hegedũs, R. Ormándi, and M. Jelasity, "Gossip-based Learning under Drifting Concepts in Fully Distributed Networks," in *IEEE SASO'12*, 2012, pp. 79–88.

[18] E. Loureiro, P. Nixon, and S. Dobson, "Decentralized and Optimal Control of Shared Resource Pools," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, no. 1, pp. article 14:1–31, April 2012.

[19] G. Picard, M.-P. Gleizes, and P. Glize, "Distributed Frequency Assignment Using Cooperative Self-Organization," in *IEEE SASO'07*, 2007.

[20] E. Jeanvoine, C. Morin, and D. Leprince, "Un protocole de découverte de ressources optimisé pour l'allocation de ressources dans les grilles," in *CFSE 2006*, octobre 2006, pp. 49–59.

[21] A. Iamnitchi, I. T. Foster, and D. Nurmi, "A Peer-to-Peer Approach to Resource Location in Grid Environments," in *IEEE HPDC-11)*, 2002, p. 419.

[22] V. Kumar, N. Leonar, and A. S. Morse, Eds., *Cooperative Control*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag, 2005, vol. 309.

[23] D. Weyns, S. Malek, and J. Andersson, "On Decentralized Self-Adaptation: Lessons from the Trenches and Challenges for the Future," in *ACM SEAMS'10*, 2010, pp. 84–93.

[24] L. Baresi, S. Guinea, and G. Tamburrelli, "Towards Decentralized Self-adaptive Component-based Systems," in *ACM SEAMS'08*, 2008, pp. 57–64.

[25] G. Mahdi, A. Gouaïch, and F. Michel, "Towards an Integrated Approach of Real-Time Coordination for Multi-agent Systems," in *KES-AMSTA'10*, ser. LNAI, vol. 6070. Springer-Verlag, 2010, pp. 253–262.

[26] C. Gerber and C. Jung, "Resource Management for Boundedly Optimal Agent Societies," in *ECAI'98*. John Wiley & Sons, 1998, pp. 1–6.

[27] S. Rustogi and M. Singh, "The Bases of Effective Coordination in Decentralized Multi-agent Systems," in *ATAL'98*, ser. LNAI, no. 1555. Springer-Verlag, 1999, pp. 149–161.

[28] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 2, pp. 156–172, mars 2008.

[29] J. R. Kok and N. Vlassis, "Collaborative Multiagent Reinforcement Learning by Payoff Propagation," *Journal of Machine Learning Research*, vol. 7, pp. 1789–1828, Decembre 2006.

[30] I. Dusparic and V. Cahill, "Research Issues in Multiple Policy Optimization Using Collaborative Reinforcement Learning," in *ACM SEAMS'07*, 2007, pp. 18:1–7.

[31] J. Dowling and S. Haridi, "Decentralized Reinforcement Learning for the Online Optimization of Distributed Systems," in *Reinforcement Learning: Theory and Applications*. I-Tech Education and Publishers, 2008, ch. 8, pp. 143–166.