



SAAM: A Self-Adaptive Aggregation Mechanism for Autonomous Management Systems

Rafik Makhoulfi, Guillaume Doyen, Grégory Bonnet, Dominique Gaïti

► To cite this version:

Rafik Makhoulfi, Guillaume Doyen, Grégory Bonnet, Dominique Gaïti. SAAM: A Self-Adaptive Aggregation Mechanism for Autonomous Management Systems. Proceedings of the 13th IEEE/IFIP Network Operations and Management Symposium, Apr 2012, Maui, United States. pp.667-670. hal-00951794

HAL Id: hal-00951794

<https://hal.science/hal-00951794>

Submitted on 25 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SAAM: a Self-Adaptive Aggregation Mechanism for Autonomous Management Systems

Rafik Makhoulfi*, Guillaume Doyen*, Grégory Bonnet[†] and Dominique Gaiiti*

*ICD/ERA, Université de Technologie de Troyes ; [†]GREYC/MAD, Université de Caen

Email: {rafik.makhoulfi, guillaume.doyen, dominique.gaiiti}@utt.fr ; gregory.bonnet@unicaen.fr

Abstract—In this paper, we propose a decentralized Self-Adaptive Aggregation Mechanism (SAAM) that adapts itself to the supporting network operational behavior by dynamically selecting the best aggregation approach to use. SAAM is based on (1) a fuzzy-based model that estimates the cost and performance of each aggregation scheme and (2) Multiple Attribute Decision Making (MADM) to make decisions on the best approach to use in this context. We validate SAAM by evaluating its fuzzy model and adaptation cost, and by comparing its utility to the one of existing situated and global schemes.

Index Terms—Autonomous Networking, Decentralized Aggregation, Management Information, Fuzzy Logic, MADM.

I. INTRODUCTION

Recently, several approaches have emerged in order to manage networks and services in an autonomous way. In this kind of frameworks, the hosting environment is continuously evolving, for example, in terms of size and dynamics. Consequently, the management approach should adapt itself to such different operational states. However, existing autonomous management systems propose to deploy a unique slightly adaptive approach to manage the network elements whatever the nature of their environment is. The same problematic is addressed in the case of aggregation schemes.

In this paper, we propose a self-adaptive management approach that adapts itself to the supporting network operational behavior by dynamically selecting the best approach to use while optimizing the management cost and performance. We especially investigate the use of this architecture in the case of aggregated information monitoring. We call this framework Self-Adaptive Aggregation Mechanism (SAAM). The latter ensures that, whatever the network operational state, the best monitoring approach is used, providing thus the most accurate aggregated information to the decision making processes. To design SAAM, we rely on fuzzy logic to estimate the cost and the performance of each aggregation scheme and also on MADM to decide what is the best aggregation approach to use. We validate SAAM by evaluating its fuzzy model and adaptation cost, and by comparing its utility to the one of existing situated and global schemes.

This paper is organized as follows. Section II discusses the related work on the existing autonomous management systems. Afterwards, Section III describes the proposed self-adaptive aggregation mechanism. Section IV is dedicated to the validation of the proposed framework. Finally, Section V concludes the paper and presents our futur work.

II. RELATED WORK

In the context of autonomous network management, few works propose a self-adaptive approaches that adapt themselves to the supporting network operational behavior.

For example, in [1], the authors propose to use a strategy-tree on a policy-based management system of a data center. They defined three strategies to process requests for ressources. The proposed mechanism aims at dynamically alter strategies, by choosing the one maximizing the cumulative weekly net profit while limiting the number of SLA (Service Level Agreement) violations. In [2], the authors present a work in progress to define a framework for self-adaptive monitoring relying on the characterization of governability, adaptability and configurability. They present the global architecture of their framework and some concepts like the capabilities levels to be managed in order to perform on line monitoring adaptation. In [3], the authors present an architecture for building adaptable autonomic managers (AM). According to the current and past behavior of the administration tasks executed by the AMs, the proposed approach dynamically decides to add a new task or, to delete or modify the current tasks.

Globally, for [1], the first validation shows that, compared to the case where strategies are executed individually, the proposed approach did not often improve profit. In [2], no details are provided on the design and validation of their mechanism. Finally, contrary to [3], in our case we are not interested in the adaptation of administration tasks, but in the one of the monitoring framework itself.

III. SELF-ADAPTIVE AGGREGATION MECHANISM

A. Considered aggregation schemes

In order to design efficient management frameworks that minimize the cost and maximize the performance of management, we propose a self-adaptive aggregation mechanism. The latter infers the current network operational state and dynamically decides which aggregation approach should be used and/or what changes must occur in it. SAAM considers typical situated and global aggregation schemes that we evaluated in [4]. These are: (1) a push-pull gossip scheme inspired from [5], where at each round, each node contacts one neighbor chosen uniformly at random and exchanges information with it ; (2) a push tree-based scheme based on GAP [6], where nodes communicate their local aggregates to a single root node that computes an overall aggregate

and spreads it on all the interested nodes through a publish-subscribe mechanism ; and finally (3) a pull-based situated scheme based on the membership protocol HyParView [7] and Propagate2All [8], where each node maintains a view of its h-hops neighbors. We consider a one-hop situated view (SV1) and a two-hops situated view (SV2).

B. SAAM Architecture

The control loop of autonomic computing on each AM consists in four main operational blocks that are Monitor, Analyze, Plan, Execute (MAPE) [9]. As depicted in Figure 1, this model can be applied to the adaptive configuration of the management plane with four components that enable a dynamically selection of a management strategy according to the current context. Thus, the resulting system operates as a meta-management system that provides an autonomous support for the management plane to select the best approach. The global architecture of the proposed system is composed of the following blocks.

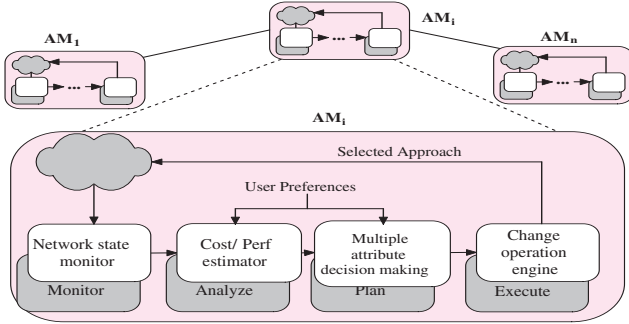


Fig. 1. Global Architecture of the Self-Adaptive Aggregation Framework

1) *Network State Monitor (NSM)*: This process is responsible for the monitoring of the managed network operational state. It uses sensors to collect data from local or from other AMs and then it sends them to the next component for analysis. In case of distributed data monitoring, these data can be collected through an aggregation scheme. The NSM collects information about the Network Dynamics (ND), Network Size (NS) and aggregated data dynamics (ID: Information Dynamics). These metrics have a direct impact on the cost and performance of aggregation [4].

2) *Cost/Perf Estimator (CPE)*: The CPE is composed of two Fuzzy Logic Controllers (FLCs) that analyze the data provided by the NSM and estimate the cost and the performance of each aggregation scheme. The CPE outputs are Convergence Time (CT), Communication Cost (CC) and Aggregation Accuracy (AA). We rely on fuzzy logic [10] because it provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. It allows a performant decision-making under incomplete and uncertain information, here, about the network operational state. The developed FLCs operates according the following main steps:

- **Fuzzification**: the CPE inputs/outputs are converted into fuzzy variables. The fuzzy sets of each variable are de-

fined according to the performance evaluation presented in [4] (Figure 2). The set of membership functions we defined for FLC variables are Very Low (VL), Low (L), Medium (M), High (H) and very high (VH).

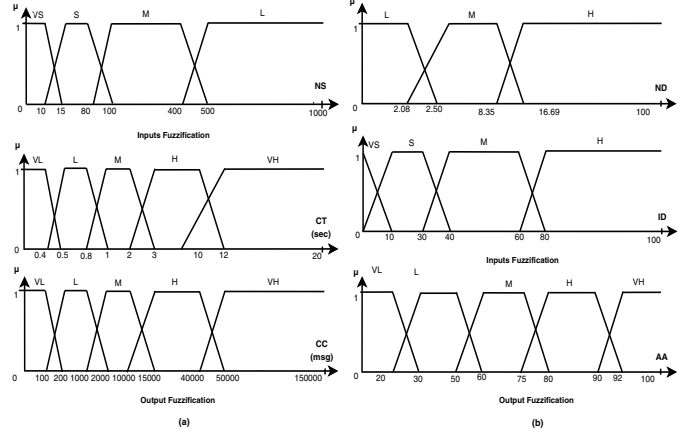


Fig. 2. Fuzzification of the FLCs inputs and outputs: (a) FLC1, (b) FLC2.

- **Inference**: according to the CPE inputs, in order to evaluate which control rules have to be activated at a given time and then decide what the output should be, each FLC relies on an inference mechanism. Tables I and II illustrate, respectively, the rules used by the FLCs to estimate both the cost and the performance of each aggregation scheme.

TABLE I
RULES OF THE COST ESTIMATOR FLC.

		Tree		Gossip		SV1		SV2	
		CT	CC	CT	CC	CT	CC	CT	CC
NS	VL	L	VL	M	VL	VL	VL	VL	VL
	L	M	VL	H	M	L	L	L	M
	M	M	L	VH	M	L	M	L	H
	H	M	M	VH	H	L	H	L	VH

TABLE II
RULES OF THE PERFORMANCE ESTIMATOR FLC.

		Tree			Gossip			SV1			SV2		
		ND	L	M	H	ND	L	M	H	ND	L	M	H
ID	VL	VH	VL	VL	VH	H	M	M	M	M	H	M	M
	L	VH	VL	VL	VH	H	M	L	L	L	M	M	M
	M	VH	VL	VL	H	H	M	L	L	L	M	M	L
	H	H	VL	VL	H	M	L	L	L	L	M	M	L

- **Defuzzification**: it converts the conclusions reached by the inference mechanism into quantifiable outputs. We consider the center of gravity method which is the most standard and widely known defuzzification one [11].

3) *Multiple Attribute Decision Making (MADM)*: According to the numerical values of the cost and the performance estimated by the FLCs, the MADM [12] components compute the utility (i.e. score) of each aggregation scheme. This component integrates the user preferences through a set of weights that define a tradeoff between cost and performance. We rely on a MADM system because it is efficient and simple to use in decision support systems. It can give managers many dimensions to consider and evaluate possible alternatives under variable degrees. In our study case, MADM enables

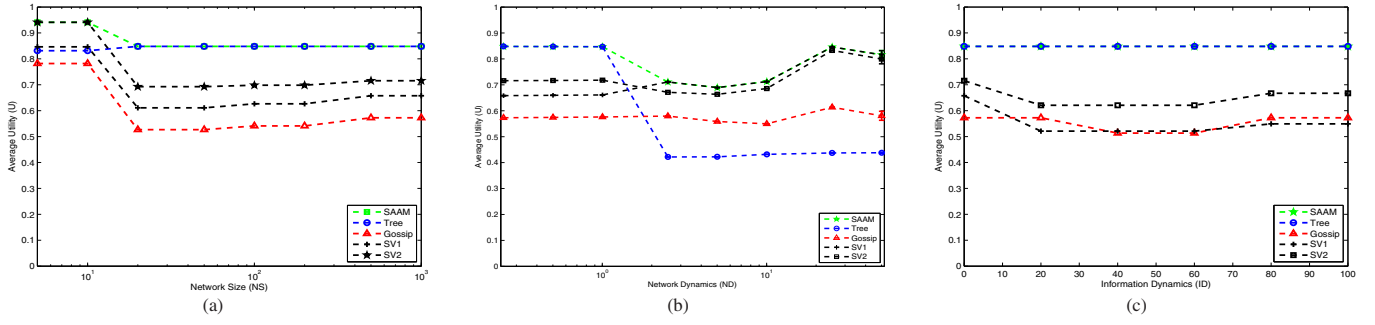


Fig. 3. Maximization of the utility according to (a) NS, (b) ND and (c) ID

a consistent ordering of aggregation approaches (alternatives) according to multiple quantitative cost and performance criteria. The MADM component is composed of:

Local utility computing component: in order to locally (on each AM) compute the utility U_i of each aggregation scheme, we use SAW (Simple Additive Weighting) [13]. The latter is a standard MADM approach. It consists of scaling the values of all attributes and summing up them for each alternative to compute an overall score. The latter is a weighted sum of all the attribute values, $U_i = w_1.CT_i + w_2.CC_i + w_3.AA_i$.

Collaborative utility computing component: in the context of decentralized monitoring, where some monitors can provide incorrect information about the network state, the AMs should collaborate in order to make consistent decisions. Each AM collaborates with other AMs and computes the average utility of each approach, by using the aggregation scheme SV1.

4) *Change Operation Engine:* This component is in charge of reconfiguring the management plane and applying the given changes. Especially, it compares the utility of the current used approach with the best one, and then decides, according to a gain threshold $GainT$, if the approach having the best utility is more relevant to be used than the current. This component can be represented through a finite state machine (FSM) with four states: Gossip (G), Tree (T), one-hop situated view (SV1) and a two-hops situated view (SV2). Since we assume that we have the same transition function between each pair of approaches, this FSM can be factorized into two abstract states $S_i, S_j \in \{G, T, SV1, SV2\}$. They represent, respectively, the current aggregation approach and the one having the best utility. The gain threshold $GainT$ enables SAAM to restrict the adaptation to only major environment changes (large $GainT$ value) or to accurately follow the environment state (low $GainT$ value).

IV. VALIDATION

A. Simulation Environment

We develop SAAM's blocs and all the considered aggregation schemes by ourselves. To evaluate the performance of SAAM, we rely on the FreePastry simulator, which enables the creation of the network nodes and the messages exchanges. Concerning the tree-based scheme, we used Scribe [4] to maintain a multicast tree for the diffusion of the global aggregate to all nodes that monitor it. Simulation parameters are illustrated in Table III. We create a network where each node is able to

execute one of the previously described aggregation schemes to compute an average aggregate of the local values on nodes. Concerning the network dynamics, we assume a Poisson process of parameter $1/\lambda$ that varies in $[2.4, 240]$ seconds for the nodes' inter-arrivals and an exponential distribution of parameter $1/\mu$, ranging in $[20, 2000]$ minutes, for the session duration. Initially, we create $\frac{\lambda}{\mu}$ nodes, and then other nodes are created according to the poisson process.

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Aggregate function	Average
Topology maintenance	60sec
Neighborhood degree	8
Max. number of nodes	1000
Adaptation round	10min
Aggregation round	1min
User preferences	$W=(0.25, 0.25, 0.5)$
	Default approach=SV1
	Ordering: churn=0 $\Rightarrow$ (SV1->SV2->Tree->Gossip) churn=1 $\Rightarrow$ (SV1->SV2->Gossip->Tree)

We validate SAAM by evaluating its capability to always instantiate the best aggregation process and by measuring its adaptation cost. We evaluate the cost and the performance of aggregation with SAAM, by computing its utility in a static environment with various network sizes and also in a dynamic context by varying the churn level and aggregated data dynamics. We evaluate the adaptation cost by measuring it in the case of a collaborative decision making (CDM) and also in the case of local decision making (LDM).

B. Results

In the following results, a 95% confidence interval is computed for each average value represented in the curves. These intervals do not appear when results are very accurate.

1) *Aggregation performance:* Given that the aggregation approach having the best utility is the one optimizing the cost and the performance of aggregation, we evaluate the performance of SAAM by periodically measuring its utility under different conditions and we compare it to those obtained with each aggregation approach executed separately. Figures 3.a, 3.b and 3.c illustrate the utility of each aggregation approach according to, respectively, network size, network dynamics and aggregated data dynamics.

We notice here that whatever the level of NS, ND or ID, SAAM outperforms the other aggregation approaches, because

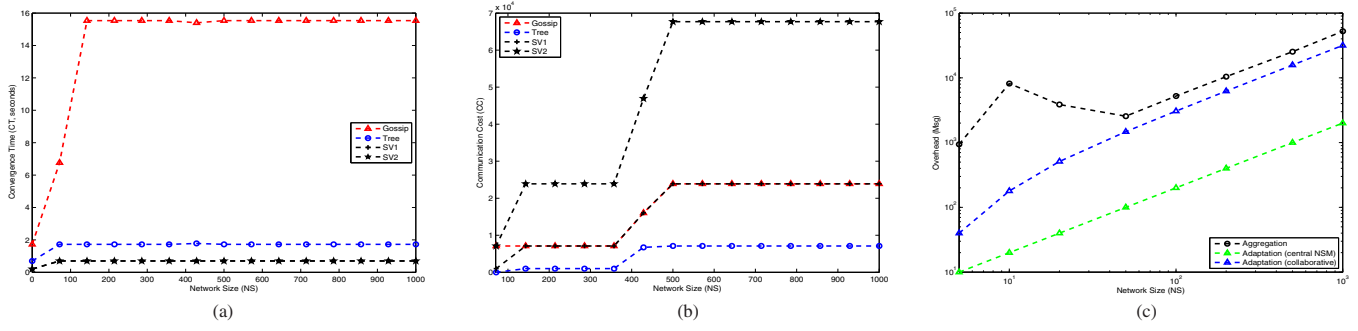


Fig. 4. Validation results on the FLCs outputs (a) CT and (b) CC, and on (c) the adaptation overhead

it periodically selects the one having the best utility. For the given user preferences, gossip often has a small utility, because it needs more time to converge than the tree or situated view. Figure 3.a shows that, globally, SAAM switches to the tree which offers a global view that is more accurate than the one of SV1. Figure 3.b shows that under a very low dynamic network, SAAM uses the tree for aggregation. But, under a higher dynamics ($ND > 2.5$), the performance of the tree is affected by this dynamics. Thus, SAAM chooses to use a situated view which better supports churn. Figure 3.c shows that SAAM always uses the tree which presents the best utility under a static network and dynamic aggregated data.

In summary, these results show that SAAM optimizes the aggregation cost and performance whatever the current context of management information and its environment.

2) *Fuzzy model validation*: We check the validity of the fuzzy model by measuring how much the distributed FLCs enable the AMs to consistently estimate by themselves, the performance of each aggregation scheme. Figures 4.a and 4.b display respectively how convergence time and communication cost depend on network size. These results are direct outputs of the cost fuzzy controller. We observe in Figure 4.a a large convergence time for gossip, followed by the tree, and SV1 et SV2. We notice in Figure 4.b that all schemes have the same distribution of the communication cost according to the network size. The communication overhead of SV2 is higher than the one of gossip and SV1, and then the tree.

In summary, these results confirm our expectations and they are consistent with those obtained through simulations in [4] showing the validity of our fuzzy model.

3) *Adaptation cost*: We measured it by the average number of messages per adaptation round needed by SAAM to adapt itself. It includes synchronization and collaboration messages. Simulation results are illustrated in Figure 4.c, where we also provide the aggregation overhead involved by SAAM. We observe here a linear distribution of the overhead according to the network size. For example, in the case of LDM and when $NS=100$, the aggregation overhead is almost 262 times higher than the adaptation overhead. This large difference is due to the fact that only some messages are needed by each node for synchronization. However, in the context of CDM, the aggregation overhead is almost twice higher than the adaptation overhead, because SAAM needs more messages

to ensure the collaboration.

V. CONCLUSION AND PERSPECTIVES

In this paper, we propose a new decentralized self-adaptive aggregation mechanism SAAM that infers the current context of the environment and dynamically decides what aggregation strategy should be used according to this context. SAAM mainly relies on fuzzy logic and MADM approaches to estimate the performance of the integrated aggregation schemes and to select the best one to use. The validation results show that SAAM efficiently optimizes the aggregation cost and performance with a relatively low adaptation cost. This work opens numerous working perspectives. We are currently performing extended simulations allowing a full featuring of its operational cost and performance. We also plan to apply our proposal to real use cases like the monitoring of complex networks such as P2P live streaming systems. In this context, we also plan to evaluate our approach in semi-real conditions through experiments over PlanetLab.

REFERENCES

- [1] B. Simmons and H. Lutfiyya, "Achieving high-level directives using strategy-trees," in *IEEE MACE*, 2009, pp. 44–57.
- [2] A. Moui and T. Desprats, "Towards self-adaptive monitoring framework for integrated management," in *IFIP AIMS PhD Workshop*, 2011, pp. 160–163.
- [3] Y. Maurel, P. Lalanda, and A. Diaconescu, "Towards introspectable, adaptable and extensible autonomic managers," in *ACM/IEEE/IFIP CNSM*, Paris, France, 2011.
- [4] R. Makhoulfi, G. Doyen, G. Bonnet, and D. Gaiti, "Impact of Dynamics on Situated and Global Aggregation Schemes," June 2011, IFIP AIMS.
- [5] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *TOCS*, vol. 23, no. 3, pp. 219–252, 2005.
- [6] M. Dam and R. Stadler, "A generic protocol for network state aggregation," in *Radiovetenskap och Kommunikation*, 2005.
- [7] J. Leita, J. Pereira, and L. Rodrigues, "Large-Scale Peer-to-Peer Autonomic Monitoring," in *DANMS*, 2008, pp. 1–5.
- [8] M. Bawa, H. Garcia-Molina, A. Gionis, and R. Motwani, "Estimating aggregates on a Peer-to-Peer network," Tech. Rep., 2003.
- [9] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.
- [10] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison Wesley Longman, 1998.
- [11] W. V. Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy Sets and Systems*, vol. 108, no. 2, pp. 159–178, 1999.
- [12] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making*. Springer-Verlag, 1981.
- [13] W. Zhang, "Handover decision using fuzzy madm in heterogeneous networks," in *IEEE Wireless Communications and Networking Conference*, vol. 2, march 2004, pp. 653–658.