Fast Root Cause Analysis on Distributed Systems by Composing Precompiled Bayesian Networks

Michał Zasadziński, Victor Muntés-Mulero, Marc Solé and David Carrera

***Abstract*— The explosion in the number of Internet of Things (IoT) and fog computing applications and the need for large data centers to host cloud and web applications make it necessary to create mechanisms to effectively control complex, heterogeneous and distributed digital ecosystems. These rapidly developing IT markets require distributed, fast and lean Root Cause Analysis (RCA) techniques to analyze dependent events. In this context, scalability and dynamically changing systems become the main obstacle to build models and infer root causes using well-established probabilistic network techniques like Bayesian Networks (BN), which are expensive to calculate and update, even when using improvements such as pre-compilation through Arithmetic Circuits (AC). In this paper, we propose a new mechanism that leverages the fact that these systems usually contain a lot of repeated elements. Our system provides a novel cache-based mechanism that, thanks to the fact that ACs can be split into subparts, will enable the reuse of previous computations to speed up the inference. The presented solution provides a fast RCA when the system model changes, without the necessity to compile the whole BN again. We evaluate our algorithm on the diagnostic system, which consists of millions of nodes, for connected IoT, fog and datacenter. Results show that the system is able to perform an order of magnitude faster, using less resources.**

***Index Terms*— arithmetic circuits, Bayesian networks, probabilistic reasoning, root cause analysis**

I. Introduction

Rapid development of the Internet of Things (IoT), and increasingly widespread use of mobile and smart devices generating frequent data collection and exchange needs are forcing organizations to change the way they engage customers, develop and deliver new products and services. Consequently, data analytics is ubiquitous, bringing intelligence to every process [1]. According to Cisco, IoT will unleash $19 trillion in new profits and cost savings globally in the next decade [2]. Besides, global data center traffic will grow nearly 3-fold from 2014 to 2019 and by 2019, more than 86 percent of workloads will be processed by cloud web services in data centers [3]. With a reference to work about Big Data and IoT frameworks in [4], data coming from IoT systems, e.g., smart cities, are characterized by high diversity of their structure, high degree of variability, high velocity and huge volume. Furthermore, data are transformed and analyzed at different layers of system, spreading from preprocessing in the sensor microprocessors to data centers running data mining and deep learning applications.

A growing amount of data and the demand for their processing bring about new approaches and paradigms in network and datacenters infrastructure. Measurements and data coming from IoT devices are not only processed in the cloud, since the infrastructure and processing capabilities can be insufficient. The needs of, e.g., geographical distribution of resources, real-time communication, incorporation with large networks are handled by fog computing. Through this paradigm, part of processing is done by edge devices or clouds closer to data sources, resulting in less latency and bandwidth usage [5].

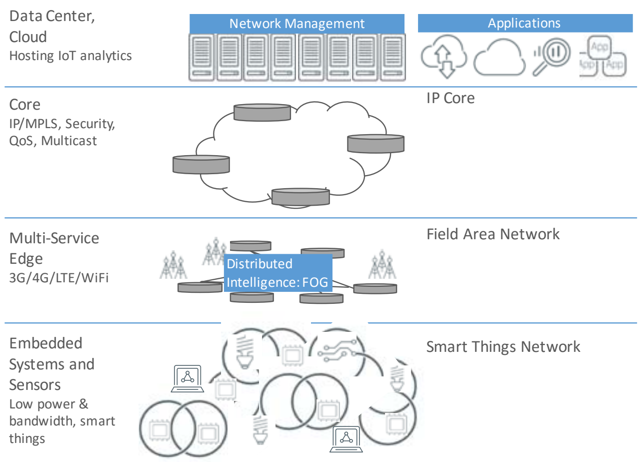

Fig. 1 - Overview of IoT computing model in reference to [6] (Permitted for use).

For an efficient monitoring, troubleshooting and management of huge IoT environments it is necessary to provide a robust Root Cause Analysis (RCA) mechanism which is scalable and tractable enough to perform fast diagnosis on the whole system and will find explanations to the problems whether they are located in the neighborhood of the particular device, other processing tier or they are compound.

Manuscript received July 1, 2016; revised July 12, 2016. This research is supported by the BigStorage project funded by European Commission (project reference 642963) through Marie Skłodowska-Curie Innovative Training Networks.

Michał Zasadziński, Victor Muntés-Mulero and Marc Solé are with CA Technologies – CA Strategic Research, Barcelona, 08940 Spain. tel: +34684052621, e-mail: {michal.zasadzinski, victor.muntes, marc.solesimo}@ca.com.

David Carrera is with The Department of Computer Architecture, Universitat Politécnica de Catalunya, Barcelona, 08034, Spain. e-mail: dcarrera@ac.upc.edu

A. Problem definition

Root Cause Analysis is one of the crucial functionalities in commercial[1,2] platforms for management and monitoring of the IoT environment. RCA based on Bayesian networks (BN) can perform an accurate diagnosis even if information about the system state is not complete like it usually happens in large and heterogeneous IoT environments extended with centralized control from datacenters and computing strategies based on fog computing. However, in order to perform more accurate root cause analysis, reasoning should be performed considering a large number of statistics, dependencies and observations, which results in (i) the large size of the diagnostic model - a network of millions of nodes and (ii) greater computational complexity. Thus, the analysis cannot be performed frequently with a satisfying level of accuracy and with the use of little resource, i.e., memory utilization at the level of hundreds of megabytes for BN with more than a million nodes. The diagnostic model needs to be changed often, e.g., when new devices are connected to the system, hardware is upgraded or the structure of the physical network changes. Using emerging technologies, such as Software Defined Infrastructure [7, 8] will cause frequent system structure changes, thus the demand to adjust the diagnostic system to it is high. Furthermore, the diagnostic system should be flexible enough for these changes, reducing recalculations as much as possible. Taking into account research in [9] on RCA using large Bayesian networks, two important conclusions can be drawn: (i) network can be divided into clusters, i.e., subnetworks which reduce calculation complexity and (ii) the root cause is usually in the region of the observed failures.

B. Contribution

We propose the method which uses global cache and it is able to reuse computations and subnetworks of the diagnostic model, which results in faster compilation – offline stage and inference – online. The reasoning is performed with use of Arithmetic Circuits (AC) which are compiled from Bayesian networks and are much faster in this process. Thanks to the use of AC, we made it possible to manipulate AC's computations and structures while the diagnosis model changes, without recompiling BNs. We are also able to reuse compiled structures for different instances of the same diagnostic model. Our contribution results in less memory footprint, faster diagnosis and better scalability of the diagnostic process compared to the use of other conventional systems, e.g., based on using a single Arithmetic Circuit for the whole system or using Case Based Reasoning (CBR), as we show later on in Section IV.

II. Related Work

A. RCA Systems

A lot of research [10-12] has been carried out to develop high-performance RCA systems in large distributed environments. However, there are not many publications proposing implementation of probabilistic reasoning for large IT systems with complex models.

A common approach for implementing root cause analysis is using classification algorithms or a specialized algorithm for alarms correlation [13]. Research in [14] aims to provide a large graph based RCA system and deploy it on the distributed servers. It is resulting with a fast and robust diagnostic system. Another example system for a complex enterprise network root cause analysis is presented in [15]. This research introduces the idea of constructing a causality graph between events in the system which is used to localize the problem. The most important issues which should be faced by an efficient RCA system based on causality and event correlation are accuracy, diagnosis time, tractability and scalability.

When compared to other diagnostic techniques [16], the Bayesian networks are distinguished as a solution for problems of an unacceptable quantity of false alarms. These alarms can be set off by a monitoring system based on a threshold approach. Moreover, the system with Bayesian reasoning is able to provide early alerts before the fault actually occurs, whereas many faults do not develop gradually over time, rather they occur instantaneously. This is not the only reason, why threshold approach is not an accurate way for the causality analysis. Another publication [17] presents large scale deployment of a diagnostic system for web applications. The solution is based on Bayesian networks and noisy-OR nodes and it uses approximate reasoning with acceptable results.

For instance, recent research on inference optimization for large scale BN is presented with the use case of failure diagnosis in Virtual Private Networks [18].

The idea of splitting Bayesian networks into objects, which are related to components, to simplify their representation is well known in literature in the area of Probabilistic Relational Models (PRM) [19, 20]. In this framework, a fairly large amount of work on structured probabilistic inference was done in [21] which produced high performance algorithms for PRM, using *d-separation*. An exact analysis and limitations of sectioning Bayesian network is fully described in [22]. Another step for optimization of Bayesian inference and model construction was made in [23], introducing general framework for canonical models. Its main objective was the simplification of complex Bayesian models, especially those in which nodes have many parents.

In [24], authors describe software health monitoring system using the BNs designed for monitoring, diagnosis and prediction in the software-hardware environment. The designed system meets the requirements of being powerful enough to reliably detect and localize significant failures with a provision of an advanced reasoning, but the research does not include large scale deployment.

Darwiche et al. [25, 26] proposed compilation to AC in order to accelerate problem resolution time using BNs. However, these techniques present numerical problems for very large systems, and furthermore, compilation requires large amounts of memory. Arithmetic circuits were successfully deployed in the diagnosis of spacecraft´s electrical systems, which is described in [27]. Another application of precompiled BN in a diagnostic system is presented in [28] but, this publication explores small

[1] http://www.striim.com/internet-of-things/
[2] http://www.splunk.com/en_us/solutions/solution-areas/internet-of-things.html

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

systems and it does not consider large scale RCA. In particular, authors neither consider replicated components nor their subgraphs in Bayesian network representation.

B. *RCA in IoT and fog computing*

However, performing such analysis considering all the tiers presented on Fig. 1 is not deeply studied in the literature. In [29] authors summarize the main research motivations for cloud services' reliability, e.g., failures and faults detection in the cloud has been hard, there has been little available research on *scalable* fault detection methods or difficulties in recognizing the faults leading to failures. Besides, Aggarwal [30] states the problem of having incomplete data transmitted from sensors and the significance of this problem for Big Data analytics. In [31] it is stated that failures in fog computing can be localized in sensors, network, lack of network coverage, service platform or the web application. Authors in [32] propose integration of Big Data with a Cyber Physical System, describe a data-driven approach to building fault tolerant control systems and they emphasize the significance. Moreover, the accurate mathematical models, will not be able to deal with the scale and computational complexity of large Internet of Things structures, thus another set of solutions is demanded.

III. Proposed Algorithm

A. *Most Probable Explanation and Arithmetic Circuits*

Discovering the root cause in the model based on BNs, it means to solve the problem of a calculation of a Most Probable Explanation (MPE), which is defined as follows.

Most Probable Explanation (MPE) - computing an MPE is a problem of finding such an explanation $\boldsymbol{W} = \boldsymbol{w}^*$, where $\boldsymbol{W}$ stands for the set of all variables considered, including those in $\boldsymbol{e}$ – given evidence, in the Bayesian network, that maximizes the conditional probability $P(\boldsymbol{w}|\boldsymbol{e})$ [33].

$$P(\boldsymbol{w}^*|\boldsymbol{e}) = \max_{w} P(\boldsymbol{w}|\boldsymbol{e}) \qquad (1)$$

The calculation of MPE is intractable and remains NP-hard, even if all variables are binary and both outdegree and indegree of the nodes is at most two [34]. The problem can be partially dealt with by precompiling the subnetworks of the replicated elements. As we show in this paper, joining them in a certain manner allows reusing computations and as a consequence an acceleration of the diagnosis of a very large system. Arithmetic circuits were introduced in [25] and they are based on the multi-linear function which can be constructed for each Bayesian network.

Multi-linear function (MLF) for Bayesian network with variables A and B is represented as follows

$$f = \sum_{b} \prod_{ba\sim b} \lambda_b \theta_{b|a} \qquad (2)$$

where λ_b denotes evidence indicators for B and $\theta_{b|a}$ stands for parameters associated with its conditional probability depending on the value of A.

An AC describes network´s probability function, in a manner which facilitates calculations during Bayesian inference. These transformations do not cause loss of diagnostic accuracy, neither sensitivity of the original model and can be evaluated much faster. Thus, the Arithmetic Circuit can be easily transformed to the *maximizer circuit*, which as mentioned at the beginning of this section is designed to calculate MPE solutions, thus instead of adding, it performs *max* operations. The complexity of the AC compilation process time as well as inference is $O(n\,exp(w))$, where n stands for the number of variables and w for the treewidth of the input Bayesian network.

B. *Method*

The presented solution leverages from the IoT system structure, which is consisted of repeated components, e.g., sensors, actuators, smart devices, servers, routers. The following example shows how to leverage repeated structures in Bayesian networks. Method follows the concept presented and proved in [18], for a Root Cause Analysis via an approximate reasoning using subnetworks of considered nodes. In Fig. 2 where is a BN that each node has two possible states B1: {*b11*, *b12*}, B2: {*b21*, *b22*} and A: {*a1*, *a2*}. These states could have any arbitrary meaning like *b11* being "B1 is working fine" and *b12* being "B1 has a problem" for instance. An AC created from this network and designed to calculate MPE can be seen on Fig. 4.

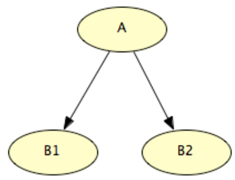

Fig. 2. The example Bayesian network to transform into AC

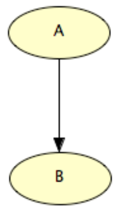

Fig. 3. The Bayesian network example

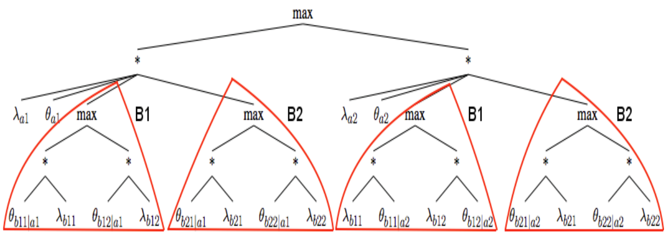

Fig. 4. The Arithmetic Circuit for the example Bayesian Network with marked parts corresponding to the B1 and B2 nodes

If states and conditional probabilities of nodes B1 and B2 are the same, there is no necessity to compile the whole BN from Fig. 2, but only consider the one shown in Fig. 3 and then aggregate computations from replicated nodes during the evaluation of the AC. The result of transformation BN from Fig. 3 into the AC can be seen on Fig. 5.

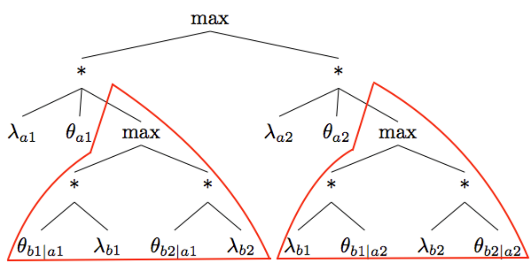

Fig. 5 – The AC with multiply and max nodes for calculating MPE in Bayesian network on Fig. 3 with marked parts corresponding to the B node

On Fig. 5 the parts of the AC that are replicated if more nodes of type B are added to the Bayesian Network from Fig. 3 are marked. It can be clearly seen that, in the worst case scenario, which is a system without replicated elements, complexity of the AC size grows as mentioned before. The system prepares an arithmetic circuit for each object type in the system. Thus, we leverage the diagnosing model by less complexity of the network. Instead of compiling the whole system, the compilation is invoked for each component class of the system. This means that if a system is composed of 1000 components of the same type, only a single compilation for the component would be required, leveraging the fact that most complex systems have a large number of replicated components.

In order to include connections between components and provide method tractability, specific Bayesian network nodes which are responsible for interconnection are cloned and placed in the referencing object (child component). This particular step is illustrated on an example network with 2 Components (

Fig. 6) that, after this processing step, results in Component 1 having one duplicated node as shown in Fig. 7.

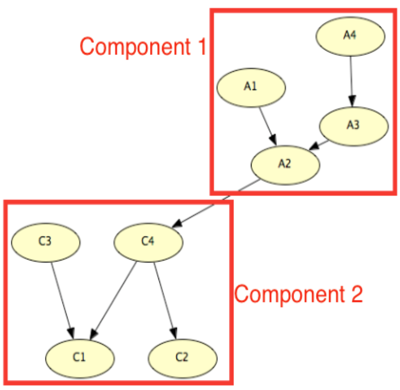

Fig. 6. Diagnostic model in form of BN, for two components of different types

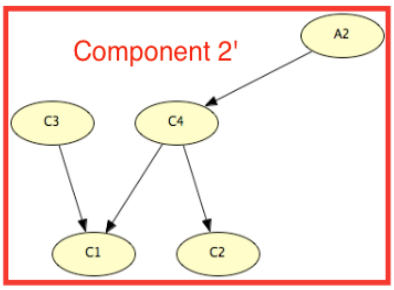

Fig. 7. BN of Component 2 after transformation, prepared for compilation into AC

Below, we present Algorithm 1 for evaluation of models and aggregating the results - performing RCA. The input consists of (i) compiled diagnostics models – AC for each component type, including a reference to the external nodes, (ii) system instances schema defining dependencies between instances' specific nodes, number of instances of each component and their connections and (iii) set of evidence e (observations of a state). As a result, marginal probabilities for each variable in the system are received.

We use the following notation:

$\boldsymbol{I}$: single instance of a component

$\boldsymbol{I.S}$: nodes in instance $\boldsymbol{I}$ which are referenced to by other external nodes from other instances

$\boldsymbol{I.P}$: nodes in instance $\boldsymbol{I}$ which are external nodes from other instances I', thus in instance $\boldsymbol{I}$ they are cloned

$\boldsymbol{I.N}$: internal nodes (including $I.S$)

$\boldsymbol{s.A}$: aggregated value of a node s, which is referenced by external nodes

$\boldsymbol{p.v}$: value of a node p

Algorithm 1: Evaluation algorithm - pseudo code

Input: Compiled models **M,** instances schema: ∏, set of evidence **e**

start with instances where **I.S = Ø**
foreach I in ∏ **do**
 foreach s in **I.S do**
  assert **s.A** aggregated all summands
 key := (type(**I**), e, weights)
 if global cache contains value for key **then**
  result := cache[key]
 else
  foreach node **s** in **I.S do**
   weight := **s.A**+log(s.**probability**)
   add s.id→**weight** to **weights** map
  result := evaluate **M[I]** with **e** and **weights**
  put **key→result** in **cache**
 print **result** for **I.N**
 foreach node **p** in **I.P do**
  let **I´** stand for an instance where is a node that **p** was cloned from
  assign values **p.v** to its referenced node **I´.S** for aggregation
  nodes **I´.S** aggregate received value incrementally with an accumulated **I´.S.A**

IV. Evaluation

The aim of the experiments, was to compare time and memory performance of the new idea with existing reasoning approaches, and validate RCA accuracy. Used diagnostic network, simulates huge IoT and datacenter environment, and is used to simulate the scale and complexity of the real environment. Thus, the references are a conventional approach, i.e., compilation of whole BN of diagnosis model into one Arithmetic Circuit and case based reasoning.

A. Implementation

Simulation of the proposed RCA was implemented with Scala 2.11.8 and Java 8, outputting a program to run in JVM. For the purposes of an efficient AC compilation and evaluation Ace 3.0[3] library was used and for CBR tests library FreeCBR[4]. Since, calculated probabilities are small orders of magnitude, i.e., 10^{-100}, it was necessary to use logarithmic calculation space, in order to avoid interrupting

[3]Automated Reasoning Group, University Of California, Ace: http://reasoning.cs.ucla.edu/ace/

[4] http://freecbr.sourceforge.net/

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the calculations by arithmetic underflow exceptions. Before invoking the appropriate code of the program, JVM warmup is performed to avoid measuring the time of JIT compilations. Experiment for each particular model was run 5 times and the presented results are the average values. The program calls Garbage Collector before each test. Experiments were run at the following configuration: SSD disk, 2.5 GHz Intel Core i7 - 4 cores, 16GB RAM on Unix based OS. Maximum JVM heap size was set to 13GB.

B. *Results*

The proposed approximate reasoning method was evaluated on the diagnosis model which is presented on Fig. 8. On this scheme, prefix of the node label indicates the component type, i.e., S stands for server, D for IoT device, E for edge device, G for global causes, R for rack. Experiments were run for the following quantity of devices: 20 servers per rack, 3 ÷ 30 racks, 600 devices of 3 different types per server, 1 edge router per 600 devices. Belief Propagation algorithm with a limit to 10 iterations was run on the Bayesian network. This part was implemented with Figaro[5] library, and the result is not presented on the plots, because the evaluation of the model for the first iteration took 2855s with a maximum memory usage of 6 GB and the offline stage lasted for 130s.

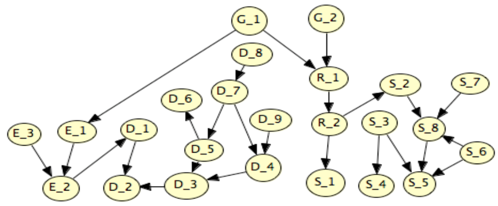

Fig. 8. Simplified Bayesian network presenting relations between events in different components. One instance of each component type is shown only.

The following plots illustrate maximum memory consumption and time of online and offline stages for the considered algorithms. Inference performance was also investigated in dimension of an evidence entropy, which was changed by setting the same or random values as the observations. Quality of the proposed cache based RCA method is presented in Table I.

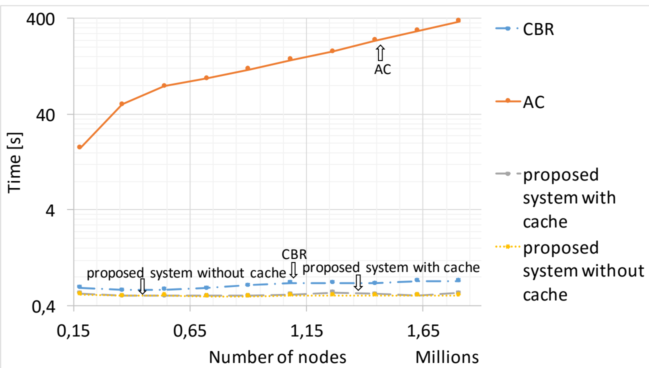

Fig. 9. Offline stage time

[5] https://github.com/p2t2/figaro

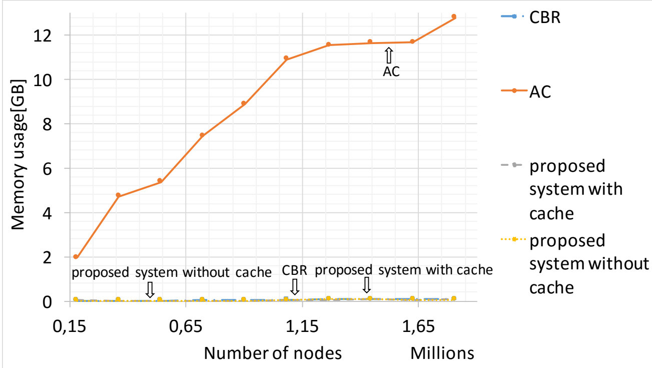

Fig. 10. Offline stage maximum memory usage

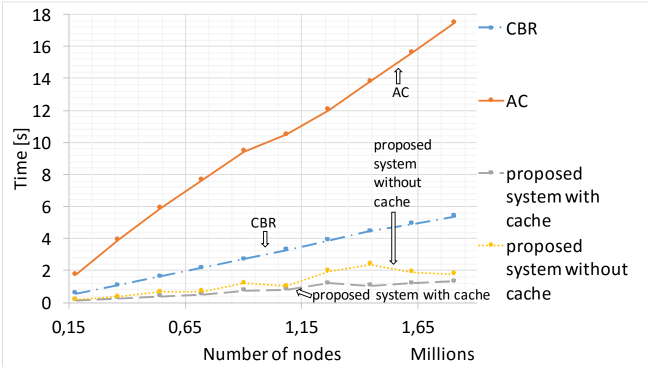

Fig. 11. Evaluation time

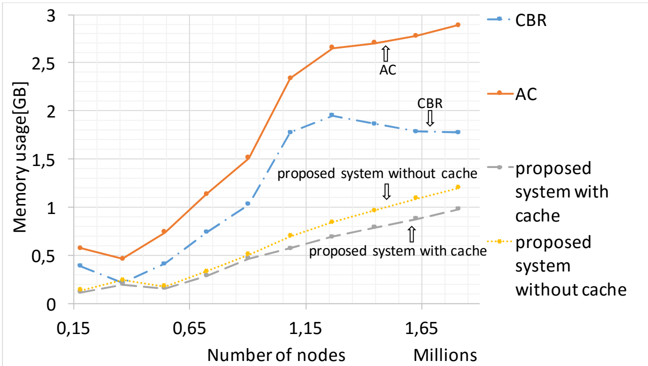

Fig. 12. Evaluation maximum memory usage

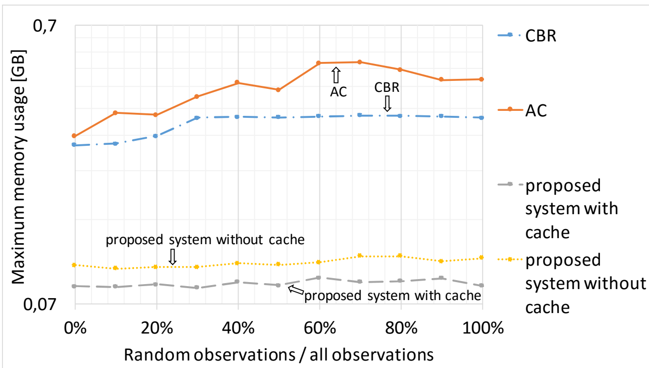

Fig. 13. Maximum memory usage depending on observations entropy

Table I - Accuracy of the proposed method, measured on the above model

Measure	Value
Sensitivity (TPR) - true positive rate	0.5726
Precision (PPV)	0.9988
Negative predictive value (NPV)	0.7273
Specificity (SPC) - true negative rate	0.9994

Summing up, results prove that the proposed method is characterized by extremely good precision and specificity,

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and acceptable enough sensitivity and negative predictive value. In addition, it is about an order of magnitude faster in evaluation and requires more than two times less memory comparing to the accurate approach with compilation the whole BN to a single AC.

V. Future Work

Further research area is focused on the deployment of the fast root cause analysis system for efficient diagnostics in Big Data systems for the smart city. The other significant step to take, is the creation of a new compilation algorithm for Bayesian networks to leverage repeated structures and improve accuracy of the method proposed in this publication. It will be achieved by more complex analysis of the nodes' dependencies between components.

References

[1] "Tech trends for 2015: the year in which the digital-first world takes hold," 2015. [Online]. Available: http://www.techradar.com/us/news/world-of-tech/tech-trends-for-2015-the-year-in-which-the-digital-first-world-takes-hold-1280750.

[2] "Cisco Adds Big Data to Its 'Internet of Everything' Push," 2014. [Online]. Available: http://www.bloomberg.com/news/2014-12-11/cisco-adds-big-data-to-its-internet-of-everything-push.html.

[3] "Cisco Global Cloud Index: Forecast and Methodology, 2014–2019," 2015. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf.

[4] M. Strohbach, H. Ziekow, V. Gazis and N. Akiva, "Towards a Big Data Analytics Framework for IoT and Smart City Applications," in *Modeling and Processing for Next-Generation Big-Data Technologies: With Applications and Case Studies*, Darmstadt, Springer, 2015, pp. 257-282.

[5] L. M. Vaquero and L. Rodero-Merino, "Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing," *ACM SIGCOMM Computer Communication Review,* vol. 44, no. 5, pp. 27-32, October 2014.

[6] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *MCC '12 Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, Helsinki, 2012.

[7] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu and A. V. Vasilakos, "Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey," *Mobile Networks and Applications,* vol. 20, pp. 4-18, 2015.

[8] T. Lin, J.-M. Kang, H. Bannazadeh and A. Leon-Garcia, "Enabling SDN applications on Software-Defined Infrastructure," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, Krakow, 2014.

[9] L. Bennacer, L. Ciavaglia, S. Ghamri-Doudane, A. Chibani, Y. Amirat and A. Mellouk, "Scalable and fast root cause analysis using inter cluster inference," in *IEEE ICC - Next-Generation Networking Symposium*, 2013.

[10] H. Nguyen, Z. Shen, Y. Tan and X. Gu, "FChain: Toward black-box online fault localization for cloud systems," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, 2013.

[11] C. Wang, K. Schwan, B. Laub, M. Kesavan and A. Gavrilovska, "Exploring Graph Analytics for Cloud Troubleshooting," in *11th International Conference on Autonomic Computing*, Philadelphia, 2014.

[12] G. Bronevetsky, I. Laguna, S. Bagchi and B. R. de Supinski, "Automatic fault characterization via abnormality-enhanced classification," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, 2012.

[13] M. Miyazawa and K. Nishimura, "Scalable Root Cause Analysis Assisted by Classified Alarm Information Model Based Algorithm," in *2011 7th International Conference on Network and Service Management*, Ohara Fujimino City, 2011.

[14] Z. Guo, D. Zhou, H. Lin, M. Yang, F. Long, C. Deng, C. Liu and L. Zhou, "G2: A Graph Processing System for Diagnosing Distributed Systems," in *Proceedings of the 2011 USENIX Annual Technical Conference (USENIX ATC '11)*, 2011.

[15] S. A. Yemini, S. Kliger and E. Mozes, "High Speed and Robust Event Correlation," *IEEE Communications Magazine,* pp. 82-90, 1996.

[16] S. P. Kavulya, K. Joshi, F. Di Giandomenico and P. Narasimhan, "Failure Diagnosis of Complex Systems," in *Resilience Assessment and Evaluation of Computing Systems*, Berlin, 2012, pp. 239-261.

[17] E. Kiciman, D. Maltz and J. C. Platt, "Fast Variational Inference for Large-scale Internet Diagnosis," in *Advances in Neural Information Processing Systems 20*, 2007.

[18] L. Bennacer, Y. Amirat, A. Chibani, A. Mellouk and L. Ciavaglia, "Self-Diagnosis Technique for Virtual Private Networks Combining Bayesian Networks and Case-Based Reasoning," *IEEE Transactions on Automation Science and Engineering,* vol. 12, no. 1, pp. 354-366, January 2015.

[19] N. Friedman, L. Getoor, D. Koller and A. Pfeffer, "Learning probabilistic relational models," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[20] L. Getoor, N. Friedman, D. Koller, A. Pfeffer and B. Taskar, "Probabilistic Relational Models," in *Introduction to Statistical Relational Learning*, MIT Press, 2007, pp. 129-174.

[21] P. H. Wuillemin and L. Torti, "Patterns discovery for efficient structured probabilistic inference," . In Proceedings of the 5th international conference on Scalable uncertainty management (SUM'11), Salem Benferhat and John Grant (Eds.). Springer-Verlag, Berlin, Heidelberg, 247-260., 2011.

[22] Y. Xiang, K. Olesen and F. Jensen, "Some Practical Issues in Modeling Diagnostic Systems with Multiply Sectioned Bayesian Networks," in *Proceedings of the Twelfth International FLAIRS Conference*, Florida, 1999.

[23] F. J. Díez and M. J. Druzdziel, "Canonical probabilistic models for knowledge engineering," Madrid, Spain, 2006.

[24] J. Shumann, T. Mbaya, O. Mengshoel, K. Pipatsrisawat, A. Srivastava, A. Choi and A. Darwiche, "Software Health Management with Bayesian Networks," *Innovations in Systems and Software Engineering,* vol. 9, no. 4, pp. 271-292, December 2013.

[25] A. Darwiche, "A Differential Approach to Inference in Bayesian Networks," *Journal of the ACM* , vol. 50, no. 3, pp. 280-305, May 2003.

[26] M. Chavira, A. Darwiche and M. Jaeger, "Compiling relational Bayesian networks for exact inference," Int. J. Approx. Reasoning 42(1-2), 2006.

[27] O. J. Mengshoel, A. Darwiche, K. Cascio, M. Chavira, S. Poll and S. Uckun, "Diagnosing faults in electrical power systems of spacecraft and aircraft," in *In Proc. of the Twentieth Innovative Applications of Artificial Intelligence, Conference (IAAI-08)*, Chicago, 2008.

[28] S. Zermani, C. Dezan, R. Euler and J. P. Diguet, "Online Inference for Adaptive Diagnosis via Arithmetic Circuit Compilation of Bayesian Networks," in *Designing with Uncertainty: Opportunities & Challenges workshop*, York, UK, 2014.

[29] H. Madsen, B. Burtschy, G. Albeanu and F. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable Fog computing," in *20th International Conference on Systems, Signals and Image Processing*, Bucharest, 2013.

[30] C. C. Aggarwal, Managing and Mining Sensor Data, Springer, 2013, pp. 396-398;9-46.

[31] A. Vahid Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh and R. Buyya, "Fog Computing: Principles, Architectures, and Applications," in *Internet of Things: Principles and Paradigms*, Massachusetts, 2016.

[32] O. Niggemann, G. Biswas, J. S. Kinnebrew, H. Khorasgani, S. Volgmann and A. Bunte, "Data-Driven Monitoring of Cyber-Physical Systems Leveraging on Big Data and the Internet-of-Things for Diagnosis and Control," in *Proceedings of the 26th International Workshop on Principles of Diagnosis*, Lemgo, 2015.

[33] J. Pearl, Probabilistic reasoning in intelligent systems: Networks of Plausible Inference, Los Angeles: Morgan Kaufman Publishers, 1988.

[34] J. Kwisthout, "Most probable explanations in Bayesian networks: Complexity and tractability," *International Journal of Approximate Reasoning,* vol. 52, no. 9, p. 1452–1469, December 2011.

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)