

doi:10.15199/48.2015.10.47

Data center resources monitoring for resilient telecommunication services

Abstract. In this article authors present a specific tool for Data Center resources monitoring (both physical and virtual), utilizing model environment of a proprietary Cloud unit. Here-in discussed are multiple aspects of resource monitoring contributing to the overall rules of resiliency measurements in the deployed Data Center environment. Depicted results of the initial test-bed system monitoring reveal properties of a Cloud-based foundation for telecommunication services.

Streszczenie. W artykule przedstawiono autorskie rozwiązanie w postaci narzędzia do monitorowania zasobów w obiektach typu Centrum Danych. W obliczu rosnącej popularności środowisk z tego rodzaju w zastosowaniach telekomunikacyjnych istotnym aspektem określania jakości usług jest monitorowanie zasobów sprzętowych. Dotyczy to zarówno warstwy fizycznej środowiska, jak również wirtualnej, która w znaczącym stopniu definiuje wydajność całego systemu. W niniejszym artykule ukazano rezultaty projektowania, wykonania i testowania narzędzia w środowisku Data Center. (**Monitorowanie zasobów Centrum Danych jako element niezawodnego środowiska dla usług telekomunikacyjnych**).

Keywords: Data Center, monitoring, resiliency, cloud computing.

Słowa kluczowe: Centrum Danych, monitorowanie, niezawodność, chmura obliczeniowa.

Introduction

In a face of an emerging Data Center (DC) environments utilization for a novel telecommunication services offering Cloud-based functionalities, the utmost importance is to create a reliable system for the deployment of the designed service [1,2]. In order to create such system in cloud computing or DC environments, the management sub-systems of those include two fundamental tasks: the first one is an active job of monitoring that keeps track of selected performance metrics in hardware and software domain, while the second is monitoring data analysis that processes those metrics for the purpose of resource provisioning, troubleshooting or other relevant management tasks. From the point of view of Cloud Service Provider (CSP), monitoring is necessary to understand where to locate the problem within the managed infrastructure [3-5], while for Consumers part of the service, to understand if any occurring issue is due to a failure on the Provider's part or is caused by other factors [6-8]. Hence the motivation to create a tool supporting aforementioned functionalities.

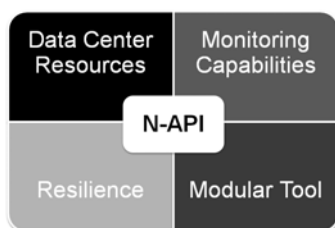


Fig.1. The functional goals of the N-API project

Discussing the research problem defined upon the following project functional goals (Fig. 1), incorporated in the designed tool are the following features:

- a modular built - to add or merge designed functionalities with the new ones, create a transparent model of interactions, and to support further expansions by structural addition of vendor specific features - all kept in an organized manner for an efficient development;
- user defined monitoring capabilities, for both physical and virtual resources of the deployed DC unit;
- support for the resiliency evaluation functionality.

Finally, in order to achieve the aforementioned attributes, the following milestones are defined:

- creation of an Application Programming Interface (API) for Data Center monitoring purpose;
- definition of a set of parameters characterizing Data Center availability and structure (i.e. provisioned processing capabilities usage, RAM memory charge, storage utilization, topology discovery);
- development of an application supporting and disposing API functionalities.

API for Data Center monitoring

Realization of the defined research goals was accomplished utilizing real-life deployment of a Cisco Data Center for telecommunication services. Designed API for monitoring purposes is utilizing the Cisco UCS Director mechanisms and functionalities integrated in the management package. As a result, deployed, programmable interface of N-API project enables users not only to monitor basic performance parameters of a Data Center, but also to perform advanced analysis of the gathered metering samples within the Analysis Extension module. Architecture of the designed N-API tool is presented on the scheme in the Fig. 2. Concerning insights to the deployed mechanisms (in JAVA/JSON environment), they are based on a Representational State Transfer (REST) functionalities. Specific connectivity sockets were designed to support both local and remote access to the Data Center monitoring features (utilizing secure API key access). The designed API enables Joint Singleton approach towards Data Center access, what means only a singular UCS instance is required for a compound operation (specification of port, protocol and the destination address is required). Connectivity scheme is presented in the Fig. 3. Delivery of selected monitoring parameters is achieved by detail, report-based resource monitoring for the following entities of the proprietary Data Center (further on called *context units*):

- *GlobalClient* - representing the compound Data Center structure; is of the highest hierarchical level; reveals architectural layout and topology of the monitored Data Center unit;
- *Cloud* - delivers information about every Cloud unit deployed in the Data Center domain; structure, topology, addressing and other specific parameters are presented;

Table 1. N-API S2 REST functionalities - available methods

Method	Parameters	Exemplary Call	Returns	Comment
getVMS	None	http://localhost:8084/napi/analysis/getVMS	{"vmlDs":["id1,id2,...,idn]}	Check the list of all available VMs of a D.C.
getCPU Analysis	vmlD timePeriod	http://localhost:8084/napi/analysis/getCPUAnalysis?vmlD=488&timePeriod=hourly	AnalysisResultCombined	vmlD has to be one of the vmlDs elements set. timePeriod has to be one of the following set: <i>hourly, daily, weekly, monthly.</i>
getMemory Analysis	vmlD timePeriod	http://localhost:8084/napi/analysis/getMemoryAnalysis?vmlD=488&timePeriod=hourly	AnalysisResultCombined	vmlD has to be one of the vmlDs elements set. timePeriod has to be one of the following set: <i>hourly, daily, weekly, monthly.</i>
getDisk Analysis	vmlD timePeriod	http://localhost:8084/napi/analysis/getDiskAnalysis?vmlD=488&timePeriod=hourly	AnalysisResultCombined	vmlD has to be one of the vmlDs elements set. timePeriod has to be one of the following set: <i>hourly, daily, weekly, monthly.</i>

- *VirtualDataCenter (vDC)* - is a subunit for the Cloud entity level that enables further partitioning of a DC resources;

- *VirtualMachine (VM)* - states the most important entity in the Data Center domain; is a basic unit for CSP offerings; performance of a single VM lays a foundation for the Service Level Agreement (SLA) over which charging and billing is conducted;

- *HostNode* and *DataStore* - reflect the physical configuration, assets and topology of the Data Center; deliver addressing information, network adapter states and storage capabilities of the monitored environment.

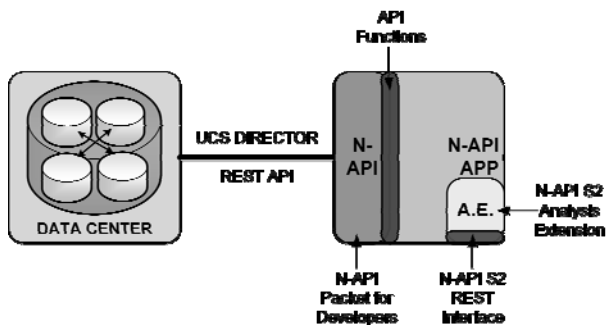


Fig.2. N-API architectural layout

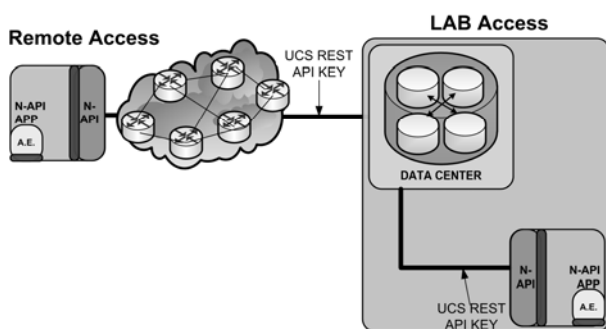


Fig.3. N-API outbound and inbound access schemes

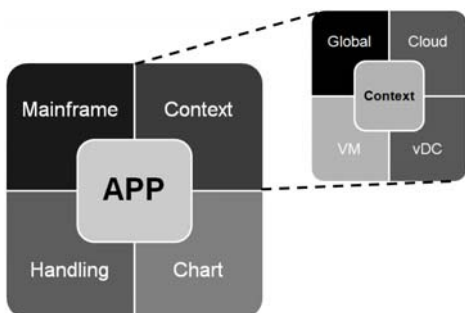


Fig.4. N-API APP functional modules

Concerning implemented REST functionality, as one can observe on the Fig. 2., there exists an expansion called N-API S2 Analysis Extension that supports monitoring application with data analysis tools. The interface disposes GET action as a basic mechanism of inquiry. Positioned in the Table 1. are the available methods, parameters and exemplary call actions. Moreover, exemplary *json* results are presented in the Table 2. In case of a general user input failure the server instance will return *http 500* status. Possible causes may include: inappropriate VM identifier (*vmlD*); *OFF State* of the inquired VM; the amount of data for analysis is too low; the selected time resolution is out of the proper range (*timePeriod*).

Table 2. N-API analysis results - returned response structures

Result Format Type	Structure
AnalysisResultCombined	{ "xyResults": [AnalysisResultXYChart], "barResults": [AnalysisResultBarChart], "pieResults": [AnalysisResultPieChart], "singleResults": [AnalysisResultSingle], "comment": "comment" }
AnalysisResultXYChart	{ "series": { "xValues": [1.1, 2.2, 3.3 ...], "yValues": [1.1, 2.2, 3.3 ...], "comment": "comment", ... }, "chartTitle": "Title", "units": "range label", "domainLabel": "label", "comment": "comment" }
AnalysisResultBarChart	{ "series": { "values": { "xValue": 1.1, "comment": "comment", ... }, "comment": "seriesName", ... }, "chartTitle": "Title", "units": "range label", "domainAxisLabel": "label", "comment": "comment" }
AnalysisResultPieChart	{ "values": { "xValue": 1.1, "comment": "bar label", ... }, "chartTitle": "Title", "units": "range label", "comment": "comment" }
AnalysisResultSingle	{ "xValue": 1.1, "comment": "comment" }

Monitoring application

Aforementioned functionalities and features of the API layer of the N-API project are utilized within developed monitoring application (N-API APP). The N-API APP is a standalone package with the integrated packet for developers (as in Fig. 2.). In order to keep the designed tool coherent within the default environment it is also based on JAVA package alongside JFreeChart plug-in and Apache Commons Math 3 v3.3 toolbox for results presentation and GUI purposes, respectively. Graph included in the Fig. 4. reveals modules spanning functionalities of the developed APP. Designed software realizes following actions and activities delivered for the end user's disposal:

- Data Center topology assessment in a form of a structured tree,
- Inventory status (configuration/state of selected unit, i.e. VM, vDC),
- Resource monitoring: CPU, RAM, Storage,
- Reporting (export data):

- units: CPU (MHz, %), RAM and Storage (GB, %),
- instant report (average value of parameter over the period of 5 min.),
- historical data resolution: hourly, daily, weekly, monthly.

Table 3. Available analysis tools with respect to the selected tab

Tab	Tool / Plot	Comment
Interpolation	Input Values	Plot of the original values of the performance indicator.
	Input Values; Chunked Values; Grouped by Counting; Operation on Chunk: Median	Plot of the median value of the chunked samples from the original set of values.
	Input Values; Loess Interpolation; Interpolation method: Local regression algorithm with the default configuration (please refer to the Apache Commons Math 3 v3.3 toolbox). Whenever interpolated values overshoot the boundary value, the cutoff below 0 and above 100 is applied.	Plot of the interpolated values of the performance indicator.
Pie Chart	Percentile Containers	Pie Chart represents the total time the VM's performance indicator spent in a specific percentile range. Containers represent intervals: 0-50%; 50-75%; 75-90%; 90-100%.
Resiliency Monitoring	Color Indicators	Visualization of the resiliency monitoring by means of distinctive color markers, representing total time spent in the appropriate percentile container. In case of a RED marker there is a Note informing about suggested activation of a chosen resiliency mechanism.
Bar Chart	Minimum	Displays the minimum value from the analyzed set of values.
	Maximum	Presents the maximum value out of the analyzed sequence.
	Arithmetic Mean	Calculates an arithmetic mean from the analyzed sequence.
	Median	Calculates a median value from the analyzed set of values.
	Standard Deviation	Calculates standard deviation value out of the considered set.

essentials of N-API Application GUI. Presented by means of Table 3. are the available analysis tools concerning actual selection of the tab (i.e. Interpolation, Pie Chart etc.). Composition presented on the Fig. 8. depicts all of the available analysis tools and their results in both textual and graphical forms.

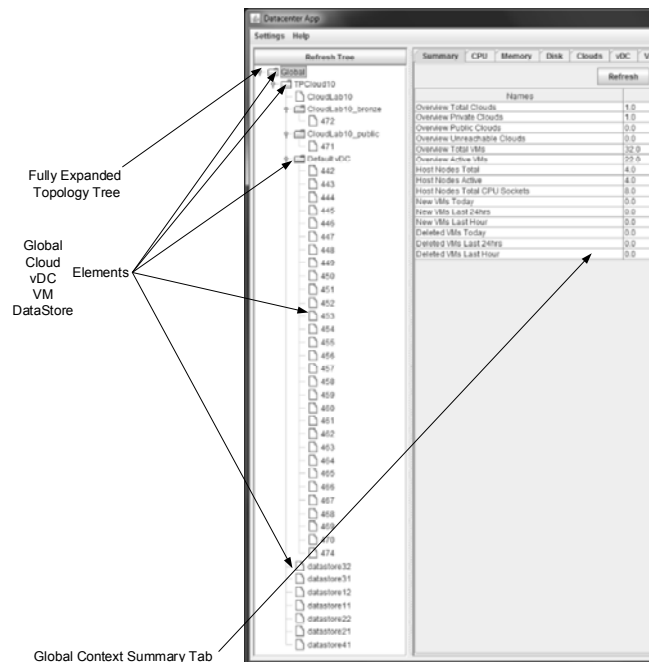


Fig.5. N-API APP discovered topology tree

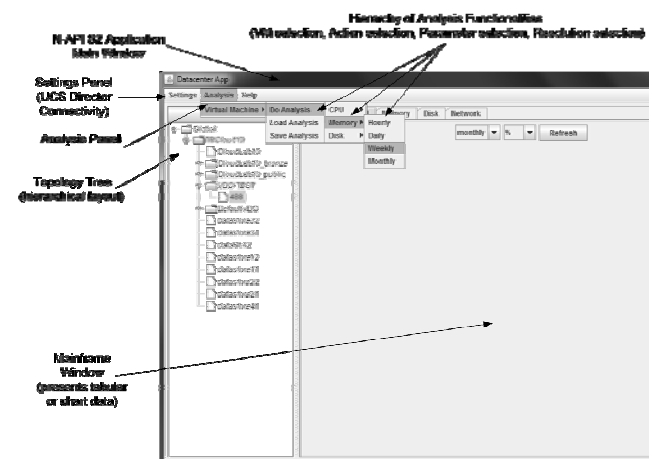


Fig.6. N-API APP Analysis Extension mainframe window

Once the application is executed connectivity module is being activated, afterwards appropriate routine is run. Whenever ready, one can select the VM-of-interest from the acquired Topology Tree, then navigate to the Analysis List from the top bar. Analysis mechanisms are performed over the VM unit, that is the lowest hierarchy element. From the *Analysis->Virtual Machine* selection one can dispose three, self-explanatory functionalities: *Do Analysis*, *Load Analysis*, *Save Analysis*. The designed Analysis Extension works on the .json formatted files, thus loading and saving of analysis results is done with such a scheme. Inside the *Do Analysis* list one can find the supported analysis of performance indicators like: CPU, Memory (RAM) and Disk (Storage/HDD/SSD). Moreover, available time resolutions for analysis are coherent with reporting time instances. Acquisition and processing of the data may take significant amount of time (whenever user hovers over and clicks the selected temporal resolution, analysis process is initiated) due to vast quantity of gathered monitoring samples. Following sequence of figures (Fig. 5. to Fig. 7.) depicts the

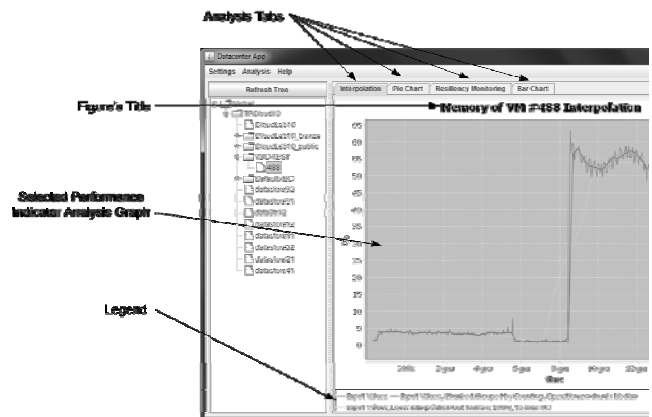


Fig.7. Interpolation analysis of RAM resources for the selected VM

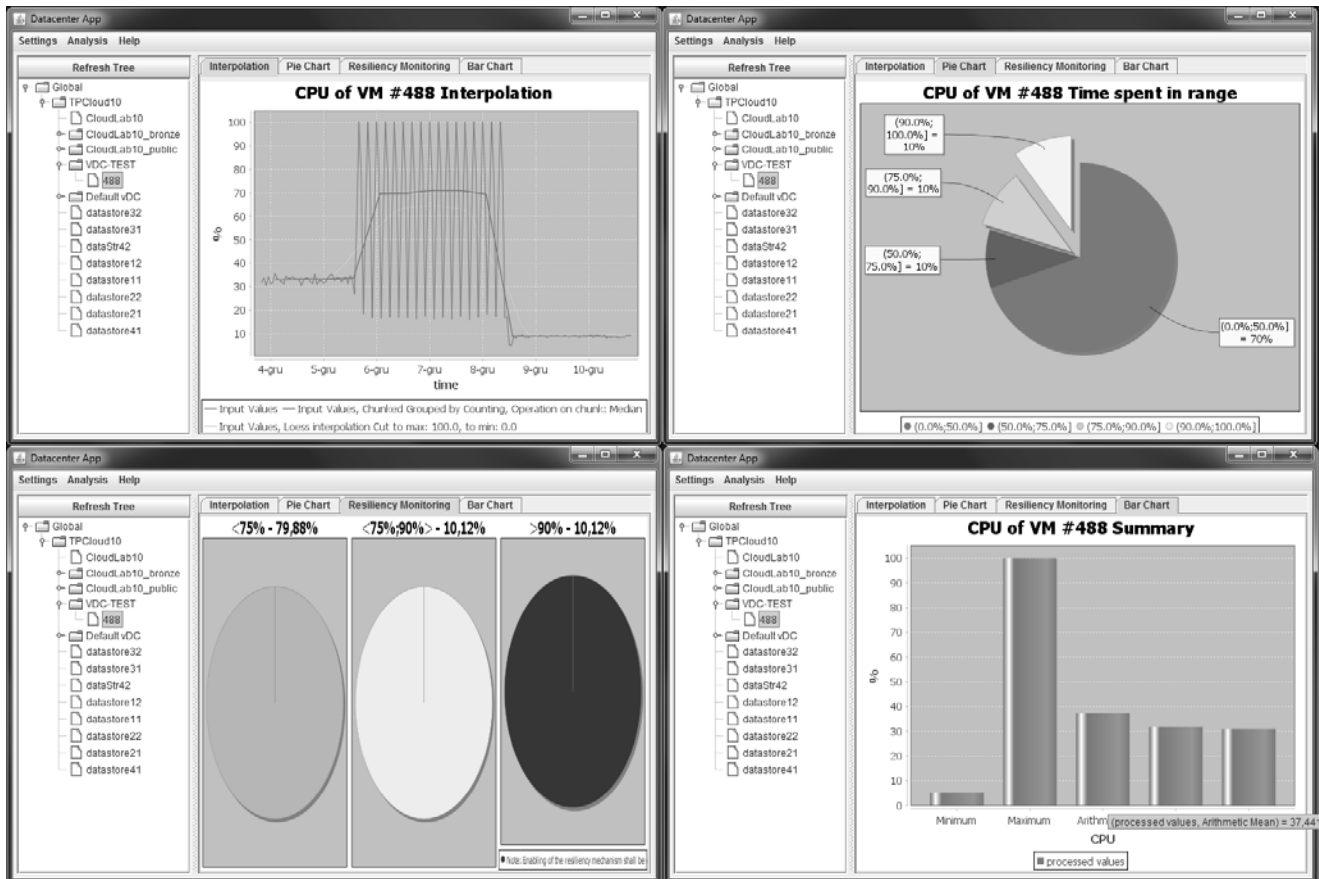


Fig.8. N-API resiliency measurement tools: *Interpolation Method* (top left), *Pie Chart* analysis (top right), *Resiliency Monitoring* (bottom left), *Bar Chart* samples analysis (bottom right).

Conclusions

Conducted deployment tests, especially designed for the purpose of controlled Data Center workload assignment (and overloading) delivered an exemplary set of data for the feed of N-API Analysis Extension module. Depicted amongst Fig. 8. are the CPU test case scenario results for the selected VM. Designed tools deliver essential pieces of information for the Cloud Service Provider to perform appropriate actions in order to keep the acclaimed SLA. Based on the acquired and analyzed data sets one can dynamically adjust performance of the considered Data Center entity (i.e. VM). Furthermore, statistical analysis of the processed samples may further improve effectiveness of the deployed DC unit.

In a final phase of deployment testing of the designed API and application features, one ought to observe an intense and crude need for resiliency evaluation capabilities. Due to heterogeneous characteristics of utilization of the Data Center resources within telecommunication services, a CSP shall perceive monitoring and resiliency analysis as a necessary condition for the successful deployment of a contemporary cloud service.

REFERENCES

- [1] Musman S., Agbolosu-Amison S., A measurable definition of resiliency using mission risk as a metric, *Mitre Technical Report*, MTR140047, McLean, VA, 2014.
- [2] Khalil Y. H., Elmaghraby A., Kumar A.: Evaluation of Resilience for Data Center Systems; *IEEE ISCC Symposium*, pp. 340-345, 2008.
- [3] Rosiński A., Dąbrowski T., Modelling reliability of uninterruptible power supply units. *Eksploracja i Niezawodność – Maintenance and Reliability*, vol.15, no. 4, 2013, pp. 409-413.

- [4] Rosiński A., Rationalisation of the maintenance process of transport telematics system comprising two types of periodic inspections; *Proceedings of the Twenty-Third International Conference on Systems Engineering - Advances in intelligent systems and computing*, vol. 1089, Springer, 2015, pp. 663-668.
- [5] Siergiejczyk M., Rosiński A., Analysis of power supply maintenance in transport telematics system. *Solid State Phenomena*, vol. 210 (2014), pp. 14-19.
- [6] Perlicki K., Siuzdak J., Dispersion Supported Transmission with a Binary Optical Signal at the Receiver. *Optical and Quantum Electronics*, 1999, vol. 31, no. 3, pp. 243-247.
- [7] Wilczewski G., Analysis of Content Quality Evaluation within 3DTV Service Distribution Systems, *National Institute of Telecommunications, Journal of Telecommunications and Information Technology (JTIT)*, vol. 1/2014, March 2014, pp. 16 - 20.
- [8] Wilczewski G., Utilization of the Software-Defined Networking Approach in a Model of a 3DTV Service, *National Institute of Telecommunications, Journal of Telecommunications and Information Technology (JTIT)*, vol. 1/2015, March 2015, pp. 32 - 36.

Authors: M.Sc. B.Sc. Grzegorz Wilczewski, Rafał Mielowski, Prof. D.Sc. Krzysztof Perlicki, Warsaw University of Technology, Institute of Telecommunications, 15/19 Nowowiejska Street, 00-665 Warsaw, E-mail: {g.wilczewski;k.perlicki}@tele.pw.edu.pl; mielowski.rafał@gmail.com; Prof. D.Sc. Mirosław Siergiejczyk, Warsaw University of Technology, Department of Transport Telecommunications, 75 Koszykowa Street, 00-662 Warsaw, E-mail: msi@wt.pw.edu.pl; Ph.D. Emil Kowalczyk, Orange Labs, Orange Polska S.A., 7 Obrzeźna Street, 02-691 Warsaw, E-mail: emil.kowalczyk@orange.com