

УДК 004.05

Padalko B.O.¹, Polska O.V.², Kudermetov R.K.³

¹ student of KNTz-615 gr. ZNTU

² senior teacher of ZNTU

³ PhD, assistant professor of ZNTU

STATE OF THE ART OF APPLICATION PERFORMANCE MANAGEMENT

In the systems development life cycle, the maintenance and operations phase is one of the most important for business and one of the longest. Depending on the source, it could be from 50% to 80% [1, 2, 3] of the overall timeline. It is a hard requirement to guarantee seamless application operation which implies detecting

anomalies rapidly, avoiding false-positive errors in feedback loops and proving a good overview of the system operation status. Having a monitoring aspect unaccounted dramatically increases deprivation chances and tends to higher outages number with irreparable damages outcome.

This document provides an overview of the state application performance management nowadays, modern problems it faces and the direction it is developing.

Application performance management (APM) is a set of software and hardware components to monitor and manage performance and availability of software applications. Modern systems based on the following older concepts:

- monitoring underlying infrastructure and resources, such as network status, bandwidth utilization, CPU usage, memory consumption, disk i/o, disks free space, etc.;
- analyzing application logs and logs of services that main application is depending on, such as data persistence systems (network file systems, databases, FTP servers), runtime caches, web servers, access control system (LDAP, SSH), but not limited to;
- tagging business transactions and collecting metrics on them;
- simulating request to an application or its components to check system status;
- managing change events such as application deployment, configuration changes, other maintenance and operations activities.

APM suites emerged as a natural response to software complexity raise in attempts to reduce operations cost and improve reliability. Compared to previous generation tools effectiveness, it was an outstanding shift forward.

The fundamental difference between APM suits and standalone tools that satisfy separate requirements of APM (which are also known as dimensions) is that APM as a methodology enables contextual analyze and background of an issue to provide a comprehensive overview of a problem or system state. Contrary, standalone instruments detect a particular problem in a chain of events at a specific application ecosystem without sufficient connection to the rest of them. The practical benefit of APM use is the reduction of the mean time to identify/detect (MTTI/MTTD) which has a direct positive impact on the mean time to resolve/recover/repair (MTTR) because of the synergy of integrated APM tools suite. The lack of interoperability [4] could make it difficult or expensive to accomplish the same with standalone tools.

Estimated that about 80% of the APM value in terms of application visibility for the business comes from Real User Monitoring (RUM) [5]. Despite many advantages, RUM has its drawbacks, most important are traffic and user interaction requirements which are barely available for non-production environments. The inconsistency of traffic volume and user behavior intricate systematic metrics

yielding with an acceptable credible interval and makes RUM to lack benchmarking capability. To eliminate these problems Synthetic Monitoring (SM or STM where T stands for Transactions) is introduced to emulate user transactions using scripted scenarios. Using real user behavior snapshots could countervail their predictability and variations limit which makes STM a cheap and affordable alternative to RUM. The best result achieved when both RUM and STM used alongside. Monitoring user transactions is the most valuable part of APM and it solves an end-user experience management problem.

Despite modern approaches, application and services log entries analysis is still the most widely used technique. It is also the oldest and cheapest approach. Logs have very poor initial visibility due to limited visualization capability without prior processing. With developing software complexity and data volume increase, they may not provide an immediate advantage at the time of an incident, though logging is a fundamental ground for historical analysis and tracing. Combined with other historical metrics further logs analytics becomes possible.

Infrastructure monitoring is often generating a large volume of metrics which with ongoing modern systems complexity increase makes it hard to analyze them without full system context. During past years relying just on infrastructure metrics has been proven to be not sufficient enough to build high-confidence automation.

In a modern complex application, performing root cause analysis is still pretty manual work even while using various APM tools at the same time. One of the possible ways to address this issue is to extend instruments to match a particular application needs and integrate better with each other. Another potential solution is to develop better classic and ML algorithms [4, 6].

With recent advances in machine learning (ML) and artificial intelligence (AI) studies utilizing APM suits enable event noise reduction, predictive event alerting, probable cause analysis, intelligent automation. This approach is known as AIOps - Artificial Intelligence for IT operations and is developing rapidly in recent years since it emerging. AIOps solves a wide range of immediate problems of immense application complexity and big data and enables further IT operations transformation. It addresses a set of applied problems, such as large logs and metrics volume processing, matching different tools output, detecting relations in tools output, alert conditions cluttering and enables building feedback loops with low false-positive errors.

We can also notice that APM vendors start offering more dimensions as a service for end users and rapidly adapt cloud-first monitoring integrating with cloud providers directly requiring little or no efforts. Consolidating more of these dimensions enables them to address interoperability problems and prepare metrics dataset for further ML analysis.

APM solved a wide range of problems, such as contextual monitoring and traceability which was very challenging to achieve with segregated tools. Further

AIOps development addresses imminent software complexity increase and brings big data management for IT operations, reduces MTI and MTTR and optimize the costs of the maintenance and operations phase even more, compared to APM impact.

REFERENCES

1. Schumaker, Carl W., Life cycle cost management: The long term view / Carl W. Schumaker, Roland D. Kankey // IEEE Proceedings of the National Aerospace and Electronics Conference, 1989, – vol. 3, – pp. 1221-1225.
2. Zarnekow, R., Distribution of cost over the application lifecycle -A multi-case study / R. Zarnekow, W. Brenner // Proceedings of the 13th European Conference on Information Systems, Information Systems in a Rapidly Changing Economy, ECIS 2005.
3. Koskinen, J. Software Maintenance Costs / J. Koskinen // 2015. Available at <https://wiki.uef.fi/display/tktWiki/Jussi+Koskinen>
4. Heger, C. Application performance management: State of the art and challenges for the future / C. Heger, A. Van Hoorn, M. Mann, D. Okanović // Proceedings of the 2017 ACM/SPEC International Conference on Performance Engineering ICPE 2017, 2017, – pp. 429-432.
5. Dragich, L. Prioritizing Gartner's APM Model. The APM Conceptual Framework / L. Dragich, – 2012. Available at <https://www.apmdigest.com/prioritizing-gartners-apm-model>
6. Ahmed T.M. Studying the effectiveness of Application Performance Management (APM) tools for detecting performance regressions for web applications: An experience report / T.M. Ahmed, C.-P. Bezemer, T.-H. Chen, A.E. Hassan, W. Shang // Proceedings - 13th Working Conference on Mining Software Repositories, MSR 2016, 2016 – pp.1-12.