**NATIONAL RESEARCH UNIVERSITY**

**HIGHER SCHOOL OF ECONOMICS**

**SCHOOL OF BUSINESS INFORMATICS**

**PROJECT PROPOSAL**

**DEVELOPMENT OF A MONITORING SYSTEM, ANALYSIS OF ANOMALIES AND TIMELY WARNING FOR MODERN IT COMPANIES**

**РАЗРАБОТКА СИСТЕМЫ ДЛЯ ПОСТРОЕНИЯ МОНИТОРИНГА,**

**АНАЛИЗА АНОМАЛИЙ И СВОЕВРЕМЕННОГО ПРЕДУПРЕЖДЕНИЯ**

**ДЛЯ СОВРЕМЕННЫХ IT КОМПАНИЙ**

Andrey Volkov, group BBI 174

Supervisor:

Sergei Gennadievich Efremov,

PhD, Associate Professor

**MOSCOW**

**2021**

This project explores the subject area of business application monitoring. It describes the main problems that modern IT companies face when building monitoring and describes the architecture of a system for monitoring business applications, analyzing anomalies and timely warning.

**Introduction**

**Background.** In the modern world, more and more companies are switching to the online business model. Behind every IT business there are hundreds of applications that support the work of modern IT companies.

However, such work cannot always be called reliable, in the modern world there are many problems associated with the availability of applications and the correct execution of their business logic.

The idea of the project is to develop a monitoring system that will export and analyze application metrics, so business will be warned of problems on time.

**Definition of terms.**

Monitoring - collection, processing, aggregation, and display in real time of quantitative indicators of the system, for example, the total number and type of requests, the number of errors and their types, the processing time of requests and the time of servers.

API - Application programming interface.

SRE - Site Reliability Engineering.

**Significance.** The solution to the problem of application unavailability is application monitoring. Monitoring allows you to profile the application at each stage of its operation, export and collect technical and business metrics of the application. Thanks to monitoring, anomaly detection and timely notification, you can achieve a transparent view of the operation of the application and respond in time to problems with its operation.

**Problem Statement.**

The ultimate goals of this project are to develop a system for building monitoring, analysis of anomalies and timely warning with the following properties:

* Software API for creating and exporting metrics.
* Deployment of cloud infrastructure for collecting and plotting graphs with metrics.
* Implementation of algorithms for analyzing anomalies in time series and export of metrics with anomalies.
* Plotting in the user interface using the software API.
* User interface for displaying graphs with application metrics.
* A warning system that sends failure notifications to popular instant messengers.

The objectives of the project are the following:

* Study the existing approaches to the implementation of monitoring in applications and existing infrastructure solutions.
* Determine the system requirements for building monitoring in terms of metrics configuration, anomaly analysis, user interface, infrastructure.
* Study algorithms for analyzing anomalies in time series.
* Develop an API for writing and configuring metrics.
* Expand infrastructure for exporting, collecting, and displaying metrics in real time.

**Scope of the Study.**

The target audience of project is any IT business that meets the following infrastructure requirements:

* Cloud architecture of application deployment.
* The ability to run and maintain docker images.
* Applications for exporting metrics must be compatible with the Kotlin (Java) programming language.

**Literature Review**

In Chapter 6 of the most famous book on application accessibility “Site Reliability Engineering: How Google Runs Production Systems” (April 26, 2016) reviewing monitoring in distributed systems, it was found that monitoring of business application is essentially important objective that solves several problems at once. Below are the given problems with the rationale for their importance.

* *Analysis of long-term trends.* E.g.: understanding how big application database is and how fast is it growing can prevent database of potential lack of available space in it.
* *Comparison with previous versions or experimental groups.* E.g.: checking is the site up and running today slower than last week can make it clear where to look for the bottleneck.
* *Alert.* E.g.: Effective alerting system can draw attention to the premises of the problem, with the aim of eliminating it.
* *Creation of information panels.* E.g.: service with decent dashboards should contain answers to the main questions about your service and be able to describe the reason of any business problem or business success.
* *Retrospective analysis for various purposes (for example, debugging).* E.g.: Retrospective analysis allows us compare different anomalies to each other and search for causation of any time. E.g.: the response time in our system has just increased; what else happened at the same time?

Besides, monitoring system also provides raw data for business intelligence and security breach analysis. Monitoring and alerting allow the system to promptly inform about an arisen or pending malfunction. If the system cannot cope with it on its own, then it is necessary for the person to analyze the received notification, determined the severity of the problem, found a workaround, and then figured out the root cause of the problem.

Despite great importance of application monitoring the actual “state of the art” leaves a lot to be desired. “Cloud applications monitoring: An industrial study” (November 2020) article published in “Information and Software Technology” journal is shedding light over the status of monitoring practices in industry, considering the following issues:

1. monitoring practices and tools adoption in industry.
2. size and complexity of industrial monitoring problems.
3. the role of software architecture and software process with respect to monitoring strategies.

They conducted mixed-methods empirical research featuring interviews and a web survey featuring 140+ practitioners from over 70 different organizations.

Even if the market provides a significant set of monitoring tools, the results of the research show a rather unattractive picture of industrial monitoring:

1. industrial decision makers do not perceive monitoring as a key asset, even if the downtime of their applications is highly correlated with the level of automation and responsiveness provided by monitoring.

monitoring is performed using crude technologies, mainly MySQL queries or similar.

1. incidents are discovered by customers and not by application owners.

All these factors lead to the conclusion that there are a lot of issues preventing people from inheriting best practices of application monitoring, which emphasizes the need of ready-made solution for monitoring infrastructure.

In Chapter 8 of “Effective Monitoring and Alerting: For Web Operations” (December 25, 2012) book the main principles of effective monitoring were underlined:

* *Get in the Habit of Measuring.* Monitoring is about extracting meaning from the data in real time. Insofar as the business meaning as well as scale of the application can change through time, it is important to continuously create and add new metrics to monitoring system.
* *Draw Conclusions Reliably.* The data at hand is not always what we expect it to be. You might wish to have gotten a different result than what it really was. That is not to say that making assumptions is inappropriate – most of the time it is necessary as it will help you save a lot of precious time. However, the more data points to support the assumption, the more reliable it will be.
* *Monitor Extensively.* Make the monitoring platform your sharpest tool. Collect metrics from all components of your application stack, ranging from network to user experience.
* *Alarm Selectively.* Alarm only on things that matter. It only makes sense to send alerts for actionable events. While any suspicious behavior should be monitored, not all of it is worth the distraction.
* *Work Smart, Not Hard.* In operations, time is a scarce and precious resource. That is one of the reasons why keeping attention only on things that matter is key.

It should be noted that it is important not only to collect metrics, but also to analyze them for the presence of anomalies. In “Time-Series Anomaly Detection Service at Microsoft” (August, 2019) article engineers from Microsoft presenting their way on anomaly detection in time series of business application metrics.

At Microsoft, it is a common need to monitor business metrics and act quickly to address the issue if there is anything outside of the normal pattern. To tackle the problem, we build a scalable system with the ability to monitor minute-level time series from various data sources. Automated diagnostic insights are provided to assist users to resolve their issues efficiently. The service has been used by more than 200 product teams within Microsoft, across Office 365, Windows, Bing and Azure organizations, with more than 4 million time-series ingested and monitored continuously.

Time-series anomaly detection is a critical module to ensure the quality of online services. An efficient, general, and accurate anomaly detection system is indispensable in real applications. Moreover, we for the first time apply the Spectral Residual (SR) model in the time-series anomaly detection task and innovatively combine the SR and CNN model to achieve an outstanding performance.

In the case of developing your own monitoring system along with the infrastructure, it is important for it to look for relevant experience in the community. In “Early-warning performance monitoring system (EPMS) using the business information of a project” (28 March 2018) article an early-warning performance monitoring system (EPMS) is proposed to objectively measure and monitor the performance of a project for early detection of inherent poor performance problems.

The EPMS is built on project progress data and consists of a business information database, an optimized theoretical model used as a baseline for measuring performance, and an index for monitoring and predicting performance. When monitoring performance with the EPMS application for a Korean construction project, it was found that the quarterly fluctuations in the index varied depending on the type of project. These results can explain the environmental changes in the project. Thus, it is expected that EPMS will become an alternative for objective performance monitoring and forecasting, while the application of existing methods is difficult due to the limited availability of data on performance indicators. Design procedures can also be useful to researchers interested in approaches to quantitative analysis of trends in various industries.

The proposed EPMS is expected to be an effective method for quantitatively monitoring project performance as it consists of a baseline performance measurement and an index using valid data, similar to existing methods such as EVMS. The proposed system is based on the contract amount, which is a leading indicator in the construction industry; therefore, it is expected to be an effective way of early detection of the accounting cliff phenomenon in the construction industry, which could have a negative impact on individual investors as well as on the national macroeconomy due to huge losses at the end of the fiscal year. despite excellent quarterly business results.

**Methods**

The architecture of system for building monitoring, analyzing anomalies and timely warning will consist of a client library with a set of modules for building metrics and analyzing anomalies and infrastructure for deploying systems for collecting metrics and a user interface. Below is a detailed description of this system and the motivation for using certain technologies.

The main technologies that will be used to implement the project:

* *Docker.* Software for automating the deployment and management of applications in containerized environments.
* *Kotlin.* JMV-based language that supports the coroutine mechanism. This allows you to cover most of the written programs in the world and use an efficient lightweight way to manage asynchronous execution.
* *Spring.* An open-source universal framework for the Java platform.
* *Kubernetes.* An open-source software for automating the deployment, scaling and management of containerized applications.

The client library will consist of the following modules:

* *Micrometer.* A module that provides a convenient and useful API for building and running metrics collection on a schedule.
* *Anomaly.* Module for analyzing anomalies in time series.
* *Grafana.* A module for plotting graphs and building dashboards in Grafana, using the “Grafana as a code” principle.

The monitoring system infrastructure will consist of the following components:

* *Business applications.* Applications that use the API of the client library mentioned above. There are two application requirements: application must be written in Java / Kotlin programming language, application must be packed in docker image.
* *Prometheus.* Storage for collecting metrics. Currently the most popular open-source project for collecting metrics in the most efficient way possible.
* *Grafana.* User interface for building dashboards and panels with metrics. Currently the most popular open-source project for building dashboards along with panels.

**Results Anticipated**

The main result of this project will be a developed system for building monitoring, anomaly analysis and timely warning. This project makes several theoretical and practical contributions. First, the findings will aid us in gaining a deeper understanding of the current situation with application monitoring in the industry, highlight the main problems and find ways to solve them. Secondly, the developed system will provide clients with useful API for building metrics and infrastructure "out of the box" for collecting, displaying, and analyzing technical and business metrics. All the source code of the project will be available in an open format and everyone can get acquainted in detail with the implementation of the mechanisms of this system.

We anticipate that the final artifacts of the system will be useful for businesses all around the world to help them build high-quality and modern monitoring in their business applications. We also anticipate community support in developing and maintaining our software product.

**Conclusion**

In order to develop a new system for building monitoring, analyzing anomalies and timely warnings, it is necessary to implement a number of points. First, research the current situation in the industry, identify bottlenecks for businesses. Secondly, to determine the set of tasks that this system will solve, to determine its limitations and to determine the set of technologies used. Thirdly, it is necessary to implement these tasks in practical terms and test the developed functionality. Fourth, collect and publish the artifacts of the system.

Within the framework of this project proposal, the goals and objectives were set, application monitoring values were identified, the subject area was investigated, the relevant literature was studied, the requirements for technologies were determined.

**References**

1. Chang-Won Kim , Wi Sung Yoo, Hyunsu Lim, Ilhan Yu, Hunhee Cho, Kyung-In Kang. (28 March 2018). Early-warning performance monitoring system (EPMS) using the business information of a project. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S026378631731253X.
2. Damian A. Tamburri, Marco Miglierina, Elisabetta Di Nitto. (November 2020). Cloud applications monitoring: An industrial study. Journal of Information and Software Technology. Retrieved from https://www.sciencedirect.com/science/article/pii/S0950584920301452.
3. Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, Qi Zhang. (August, 2019). Time-Series Anomaly Detection Service at Microsoft. Retrieved from https://dl.acm.org/doi/abs/10.1145/3292500.3330680.
4. Niall Richard Murphy, Betsy Beyer, Chris Jones, Jennifer Petoff. (April 26, 2016). Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media; 1st edition
5. Slawek Ligus. (December 25, 2012). Effective Monitoring and Alerting: For Web Operations. O'Reilly Media; 1st edition