METU, Department Of Computer Engineering

Graduation Project
Proposal Form

Project Information

Title

Automated Monitoring Solutions

Target

Public [X]	Restricted []

Proposer Information

Name(s)	Ebru AYDIN GOL
E-Mail(s)	ebru@ceng.metu.edu.tr

IP (Intellectual Property) Information

The final products will be open source under a public license.

Project Description and Background Information

Description

Various services are provided via web-servers, and users access those via browsers or mobile apps. Monitoring and alerting are fundamental techniques for guaranteeing reliability of such services. The companies developing and maintaining such services used by millions of people have custom infrastructure and experienced teams for guaranteeing reliability. However, smaller/newer companies (start ups), individuals or teams developing new systems, cannot afford to develop such infrastructure or hire specialized teams. The proposed project aims at developing **a framework** that will offer **basic monitoring and alerting** to any application running on a Unix server, and **a test platform** for the developed framework.

The team will design and implement a system that will
- **identify and gather basic metrics** of the monitored system from the operating system,
- **define log file structure** and **gather special metrics from such structured log files (log-servers)**,
- **process the gathered data**,
- **generate dashboards on web and mobile applications**.
- **generate alerts** (e-mail/sms) when a monitored metric is out of the allowed range.

In addition, the team will develop a test platform for illustrating the use of the monitoring and alerting system. The test platform will be composed of web servers that generate pre-defined metrics loaded from profiles and messages. Such metrics will be defined during the project. An example scenario is that the test server will log the time and its ID at every cycle and sleep for X seconds, which can be a random number. The monitoring and alerting system will show the cycle duration on plot and alert if it is higher than a threshold for the last 10 cycles. The wait time, X, or its distribution can be changed via the profile or the messages. The additional example metrics that can be provided by the test server include the number of requests in its service queue and the process time.

Similar Products/Projects

Stackdriver monitoring developed by Google [1]. Stackdriver collects metrics, events and metadata from servers running on Google Cloud Platform and Amazon Web Services.

Amazon CloudWatch provide monitoring and alerting services for applications running on Amazon Web Services [2].

Justification of the proposal

The project aims at developing a framework for monitoring small-scale systems and a test-pad for such monitoring applications. Many established companies develop a custom infrastructure for guaranteeing reliability and have specialized teams for improving and running such infrastructures (operations/reliability teams). However, small companies cannot afford developing their own infrastructure or employing specialized teams. Recently, the cloud platforms started to provide such services for applications running on their particular environments [1,2] with additional costs. This project aims at developing monitoring and alerting technology that can be used in Unix platforms, which will be useful for startups for individuals/teams. Furthermore, the tools developed in this project, test-pad and monitoring systems (particularly the dashboards) will be used in the future for improving the alerting mechanisms.

Contributions, Innovation and Originality Aspects of the Project

An easy to use, open sourced monitoring and alerting system that can run on a local machine or server is not available to the best of the proposers knowledge. In addition, the monitoring services provided by the cloud platforms does not have special mobile applications.
The project will allow for further research on optimizing the alerting mechanisms (the metrics, alert indications, thresholds, durations, etc.). In particular, the developed framework will be used to find the optimal alerting functions for particular metrics, to understand the relations between different metrics and system behaviors, and eventually to improve the monitoring.
As a part of the project, the team will analyze the existing monitoring solutions (e.g. [1,2]), and propose and implement distinguishing features. Note that additional features are not necessary, but it would improve the resulting project.
The team will gain experience on the problems that a member of DevOps/reliability team faces.

Technical Aspects of the Project

- Web-server development (test servers, exporting metrics, logging mechanisms)
- Web-application development (for monitoring, should show the dashboards)
- Application implementing the alerting mechanism (e-mail/sms)
- Developing a back-end server that handles data gathering and processing (reads logs, storage).
- Identifying standard metrics that can be imported for any* server and ways to import those
- Mobile application development.

Targeted Output, Targeted User/Domain Profile

The targeted output is a monitoring and alerting framework. Users running applications will have access to the dashboards via mobile and web-apps. The backend of the framework will include data gathering/processing servers and storage servers.

Web based dashboard showing the simple metrics (cpu, memory usage) is the minimal success criteria.

Project Development Environment

Unix servers, mobile applications (android only), database for data storage.

External Support

None

References

[1] Google Stackdriver Monitoring. https://cloud.google.com/monitoring
[2] Amazon CloudWatch https://aws.amazon.com/cloudwatch/