

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра технологий программирования

Аннотация к дипломной работе
«Реализация распределённой системы, управляемой событиями»

Желтовский Кирилл Дмитриевич

Научный руководитель — ст. преподаватель кафедры технологий
программирования Давидовская М. И.

Минск, 2021

Реферат

Дипломная работа, 62 с., 29 рис., 9 приложений, 7 источников.

РАСПРЕДЕЛЁННЫЕ СИСТЕМЫ, ПОТОКОВАЯ ОБРАБОТКА ДАННЫХ, МИКРОСЕРВИСНАЯ АРХИТЕКТУРА, АПАСНЕ КАФКА, JAVA, PROTOBUF, ZOOKEEPER, gRPC.

Объект исследования — объектом исследования являются высоконагруженные распределённые системы и технологии, используемые для их разработки и поддержки.

Цель работы — рассмотреть основные понятия, ключевые принципы, алгоритмы и технологии, используемые для построения распределённых систем, управляемых событиями, разработать распределённую систему, управляемую событиями, предназначенную для анализа активности разработчиков.

Со стремительным развитием информационных технологий программные продукты неизбежно усложняются, стремясь удовлетворять появляющиеся желания и требования пользователей. А значит повышаются и усложняются требования к соответствующим программным системам. Разработчики встают перед необходимостью поиска и применения решений, которые будут надёжно работать в условиях высокой нагрузки, больших потоков данных, при этом позволяя эффективно вести разработку нового функционала и поддерживать уже существующий.

Разъединение связанных сервисов с использованием потоков событий и использование модели «издатель-подписчик» оказывается довольно эффективным решением, хорошо показывающим себя на практике в условиях высоконагруженных систем. Такое решение является вполне естественным. Пользователи взаимодействуют некоторым образом с системой, нажимая на кнопки, или, скажем, переходя по ссылкам. Всё это является примером событий, происходящих в реальной жизни, и оказывается, что эти и подобные

этим события хорошо переносятся на концепцию событий в контексте программных систем.

Разрабатываемая распределённая система, управляемая событиями, служит предметом исследования и примером того, как можно, нужно или не стоит разрабатывать распределённые системы. Система предназначена для анализа активности разработчиков по их действиям на платформах хостинга кода, к примеру GitHub. Данная система может найти применение в компаниях, занимающихся разработкой ПО и желающих измерять и улучшать эффективность разработчиков.

Рэферат

Дыпломная работа, 62 ст., 29 мал., 9 дадаткаў, 7 крыніц.

РАЗМЕРКАВАНЫЯ СІСТЭМЫ, СТРУМЕНЕВАЯ АПРАЦОЎКА
ДАДЗЕННЫХ, МІКРАСЭРВІСНАЯ АРХІТЭКТУРА, АРАСНЕ KAFKA, JAVA,
PROTOBUF, ZOOKEEPER, gRPC.

Аб'ект даследавання — аб'ектам даследавання з'яўляюцца высоکانагружаныя размеркаваныя сістэмы і тэхналогіі, якія выкарыстоўваюцца для іх распрацоўкі і падтрымкі. У якасці прадмета даследавання выбіраем праектаванне, распрацоўку і даследаванне характарыстык размеркаванай сістэмы для аналізу актыўнасці распрацоўшчыкаў.

Мэты працы — разгледзець асноўныя паняцці, ключавыя прынцыпы, алгарытмы і кампрамісы, якія ўзнікаюць пры распрацоўцы размеркаваных сістэм, тэхналогіі, якія выкарыстоўваюцца для пабудовы дадзеных сістэм, разгледзець асноўныя канцэпцыі, якія ляжаць у аснове Apache Kafka, а таксама спраектаваць размеркаваную сістэму, кіруемую падзеямі, прызначаную для аналізу актыўнасці распрацоўшчыкаў.

З імклівым развіццём інфармацыйных тэхналогій праграмныя прадукты непазбежна ўскладняюцца, імкнучыся задавальняць жадання і патрабаванні карыстальнікаў. А значыць павышаюцца і ўскладняюцца патрабаванні да адпаведных праграмным сістэмах. Распрацоўшчыкі ўстаюць перад неабходнасцю пошуку і прымянення рашэнняў, якія будуць надзейна працаваць ва ўмовах высокай нагрузкі, вялікіх патокаў дадзеных, пры гэтым дазваляючы эфектыўна весці распрацоўку новага функцыяналу і падтрымліваць ўжо існуючы.

Раз'яднанне звязаных сэрвісаў з выкарыстаннем патокаў падзей і выкарыстанне мадэлі «выдавец-падпісант» аказваецца даволі эфектыўным рашэннем, добра паказвалым сябе на практыцы ва ўмовах высоکانагружанных сістэм. Такое рашэнне з'яўляецца цалкам натуральным. Карыстальнікі

ўзаемадзейнічаюць нейкім чынам з сістэмай, націскаючы на кнопкі, або, скажам, пераходзячы па спасылках. Усё гэта з'яўляецца прыкладам падзей, якія адбываюцца ў рэальным жыцці, і аказваецца, што гэтыя і падобныя гэтым падзеі добра пераносяцца на канцэпцыю падзей у кантэксце праграмных сістэм.

Распрацоўваемая размеркаваная сістэма, кіраваная падзеямі, служыць прадметам даследавання і прыкладам таго, як можна, трэба ці не варта распрацоўваць размеркаваныя сістэмы. Сістэма прызначана для аналізу актыўнасці распрацоўшчыкаў па іх дзеянням на платформах хостынгу кода, да прыкладу GitHub. Дадзеная сістэма можа знайсці прымяненне ў кампаніях, якія займаюцца распрацоўкай ПА і жадаючых вымяраць і паляпшаць эфектыўнасць распрацоўшчыкаў.

Abstract

Diploma thesis, 62 p., 29 figures, 9 appendixes, 7 sources.

DISTRIBUTED SYSTEMS, STREAMING DATA PROCESSING, MICROSERVICE ARCHITECTURE, APACHE KAFKA, JAVA, PROTOBUF, ZOOKEEPER, gRPC.

Object of research — data-intensive distributed systems and technologies used to develop and maintain these systems. As the subject of study, we choose the design, development and analysis of characteristics of the distributed system for analyzing developer's activity.

Purpose — consider general distributed systems concepts, key concepts and algorithms, tools for building distributed systems, to study Apache Kafka concepts and to design a distributed system that analyzes developer's activity.

With the rapid development of information technology, software products inevitably become more complex, striving to satisfy the emerging desires and requirements of users. This means that the requirements for the corresponding software systems are increasing and becoming more complicated. Developers are faced with the need to find and apply solutions that will reliably work under high load conditions, large data streams, while allowing effective development of new functionality and support the existing one.

Disconnecting related services using streams of events and using the “publisher-subscriber” model turns out to be a fairly effective solution that works well in practice in high-load systems. This decision is quite natural. Users interact in some way with the system by clicking on buttons, or, say, following links. These are all examples of real-life events, and it turns out that these and similar events translate well into the concept of events in the context of software systems.

The developed distributed event-driven system is the subject of research and an example of how to build a distributed system. The system is designed to analyze the

activity of developers by their actions on code hosting platforms, for example, GitHub. This system can find application in software development companies that want to measure and improve the effectiveness of developers.