МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»
(НИЯУ МИФИ)

УТВЕРЖДАЮ
Первый проректор НИЯУ МИФИ
_____ Нагорнов О.В.
«_____»_____ 2020 г.

АННОТАЦИЯ
программы повышения квалификации
«Большие данные»
(18 часов)

Разработчик:
Ровнягин М.М., к.т.н, доцент

Москва, 2020 г.

ОБЩИЕ ПОЛОЖЕНИЯ

Программа дополнительного профессионального образования по направлению «Большие данные» создана в целях повышения профессионального мастерства учителей школ-участников и школ-кандидатов проекта «ИТ-класс в московской школе». Также, данная программа может быть интересна учителям общего, среднего-профессионального и дополнительного образования, задействованным в преподавании информационных технологий в учебных заведениях, а также учителям, курирующим образовательные кружки.

Данная учебная программа позволяет получить и закрепить теоретические и практические знания в области технологий обработки больших массивов данных. Слушатели, полностью выполнившие Учебную программу и успешно прошедшие контроль знаний, получают удостоверение о повышении квалификации установленного образца.

Учебная программа посвящена изучению современных технологий обработки, хранения для BigData. Учебная программа позволит овладеть базовыми знаниями и навыками построения систем обработки больших данных включая создание высокопроизводительных систем хранения, пакетной и поточной обработки данных, системы сборки и версионирования.

ОРГАНИЗАЦИЯ УЧЕБНОГО ПРОЦЕССА

Объем учебной программы: 18 академических часов.

Форма обучения: очная.

По каждому разделу проводятся следующие виды аудиторных занятий: лекции, практические занятия и контроль знаний.

Контроль знаний проводится в форме тестирования и (или) собеседования.

СТРУКТУРА УЧЕБНОЙ ПРОГРАММЫ

Учебная программа состоит из следующих разделов:

№ п/п	Наименование разделов	Всего часов	В том числе		
			Лекции	Практика	Контроль знаний
1	Занятие 1. Современные BigData -решения и архитектуры	2	2	-	-

2	Занятие 2. Системы разработки, сборки и доставки кода	2	1	1	-
3	Занятие 3. Технология обработки и хранения данных Hadoop	2	2	-	-
4	Занятие 4. Организация хранения данных в BigData-системах	2	1	-	1
5	Занятие 5. Системы пакетной обработки данных	4	2	2	-
6	Занятие 6. Системы поточной обработки данных	2	2	-	-
7	Занятие 7. Архитектура облачных BigData-приложений	2	2	-	-
8	Занятие 8. Способы повышения производительности BigData-систем	2	1	-	1
	Всего	**18**	**13**	**3**	**2**

СОДЕРЖАНИЕ РАЗДЕЛОВ ПРОГРАММЫ

Занятие 1. Современные BigData -решения и архитектуры

Определение понятия Большие Данные. Специфика проектирования и создания систем работы с большими данными. Виды задач, решаемых BigData-системами. Зависимость способа решения задачи от ее разновидности. Составные части BigData-систем. Проблема оценки качества BigData-систем. Функциональные и нефункциональные требования. Понятия времени отклика и пропускной способности. Методы и средства оценки производительности и утилизации в системах обработки больших объемов данных. Виды обработки: синхронная и асинхронная, пакетная и поточная. Преимущества и недостатки. Соотношение видов обработки с пользовательским опытом. Системы оперативного и долгосрочного хранения данных.

Занятие 2. Системы разработки, сборки и доставки кода

Качества языка программирования, влияющие на производительность. Параллелизм, удобство использования (синтаксис), обработка ошибок, безопасность. Сравнение языков программирования на примере реализации однотипной BigData-задачи

Способы обеспечения одновременной разработки проекта несколькими людьми. Системы контроля версий. Git, SVN. Отличия. Типовые операции.

Проблема управления версиями программных модулей и публикации артефактов. Системы сборки Maven и Gradle. Репозиторий артефактов программного кода.

Специфика развертывания распределенных высокопроизводительных приложений. Введение в облачные сервисы (IaaS, PaaS, SaaS). Философия DevOps, непрерывная интеграция и доставка кода.

Практическое занятие в формате лабораторной работы

Занятие 3. Технология обработки и хранения данных Hadoop

Появление и развитие технологии Hadoop. Спектр решаемых задач. Примеры использования в крупных копаниях. Составные части Hadoop-кластера. Экосистема Hadoop-проектов.

Распределенное выполнение программ. Операции отображения и свертки. Примеры. Возможность переопределять части MapReduce программы в каркасе Hadoop.

Особенности реализации функций отображения и свертки в Java. Работа с консолью Hadoop-кластера. Спецификация параметров запуска MapReduce Java-приложения.

Возможности Hadoop Streaming API. Примеры реализации MapReduce-программ на языке python и запуска из консоли bash.

Занятие 4. Организация хранения данных в BigData-системах

Определение Базы Данных и Системы Управления Базами Данных. Организация хранения информации в классических СУБД на примере PostgreSQL. Вопросы обеспечения целостности данных. Требования ACID.

Подходы к хранению данных. Специфика хранения файлов. Архитектура популярных распределенных файловых систем на примере HDFS, NFS v4.1, Lustre. Использование распределенных файловых систем в задачах обработки больших объемов данных.

Специфика хранения данных по записям. Проблематика обеспечения согласованности данных. Теорема CAP. Архитектура NoSQL (not only SQL) на примере Apache Cassandra и их отличие от классических СУБД.

Постановка задачи на проектирование распределенной масштабируемой системы. Помощь в реализации и проверке выполнения функциональных и нефункциональных требований.

Контроль знаний в форме теста с вариантами ответов. Разбор ответов.

Занятие 5. Системы пакетной обработки данных

Состав и назначение Spark-кластера. Спектр решаемых задач. Примеры работы с источниками данных. План выполнения вычислений. Работа с основной и внешней памятью. Использование программного интерфейса Spark RDD. Состав и назначение кластера Hive. Связь с технологией Hadoop. Синтаксис запросов HiveQL. Возможность создания пользовательских функций. Предпосылки появления программного интерфейса Spark SQL. Использование HiveContext. Умные источники данных для Spark. Проблема импорта данных в системы пакетной обработки. Импорт данных в HDFS при помощи Sqoop. Настройка количества операций отображения. Передача пароля через командную строку.

Практическое занятие в формате лабораторной работы

Занятие 6. Системы поточной обработки данных

Архитектура событийно-ориентированных BigData-систем. Задача брокера сообщений. Типовые архитектуры систем передачи сообщений. Достоинства и недостатки. Семантики доставки (точно один раз, как минимум один раз, максимум один раз).

Брокер сообщений Apache Kafka. Архитектура. Производители, потребители, брокеры. Вопросы масштабирования. Принципы организации долговременного хранения сообщений. Настройка дедупликации сообщений.

Архитектура Flume. Агент, канал передачи, селектор, слив. Варианты конфигурирования.

Apache Spark Streaming. Принцип микропакетной обработки. Плавающее окно. Спектр решаемых задач. Работа с программным интерфейсом DStream.

Занятие 7. Архитектура облачных BigData-приложений

Технология Docker. Разграничение пространств процессов в операционной системе при помощи cgroups. Состав технологии: файлы докер, реестр образов, контейнеры, демон Docker. Слои доступа к файлам. Кластер Docker-хостов. Создание кластера. Масштабирование контейнеров. Маршрутизация входящих соединений. Конфигурирование развертывания в yml-файлах. Оркестрация сервисов масштабных приложений. Проблемы и вызовы. Состав кластера Kubernetes. Основные элементы развертывания (Pod, Service, Route).

Создание динамических сетей маршрутизации данных между сервисами. Отделение слоя доставки данных и авторизации от бизнес-логики.

Занятие 8. Способы повышения производительности BigData-систем

Понятие и назначение кеша данных. Ограничения. Стратегии замещения данных в кеш, стратегии предвыборки. Популярные системы кеширования для BigData-задач.

Технологии вычислений в памяти и хранения данных в памяти. Кластер Ignite. Возможности Ignite для организации кеширования, вычислений, передачи сообщений и машинного обучения на основе больших данных.

Адаптация ресурсов приложения в зависимости от нагрузки. Способы мониторинга нагрузки. Встроенная поддержка правил автоматического масштабирования в Kubernetes.

Разработке высокопроизводительного приложения на программном каркасе Apache Spark – средства и подходы.

Контроль знаний в форме теста с вариантами ответов. Разбор ответов.

Заключительное слово. Ответы на вопросы. Заполнение анкет. Сбор пожеланий.