

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э.
БАУМАНА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

«Анализ методов распределенных вычислений в распределенных системах хранения информации»

Студент: Романов Семен Константинович

Группа: ИУ7-75Б

Научный руководитель: Бекасов Денис Евгеньевич

Цель и задачи

Цель — классифицировать существующие методы распределенных вычислений.

Задачи:

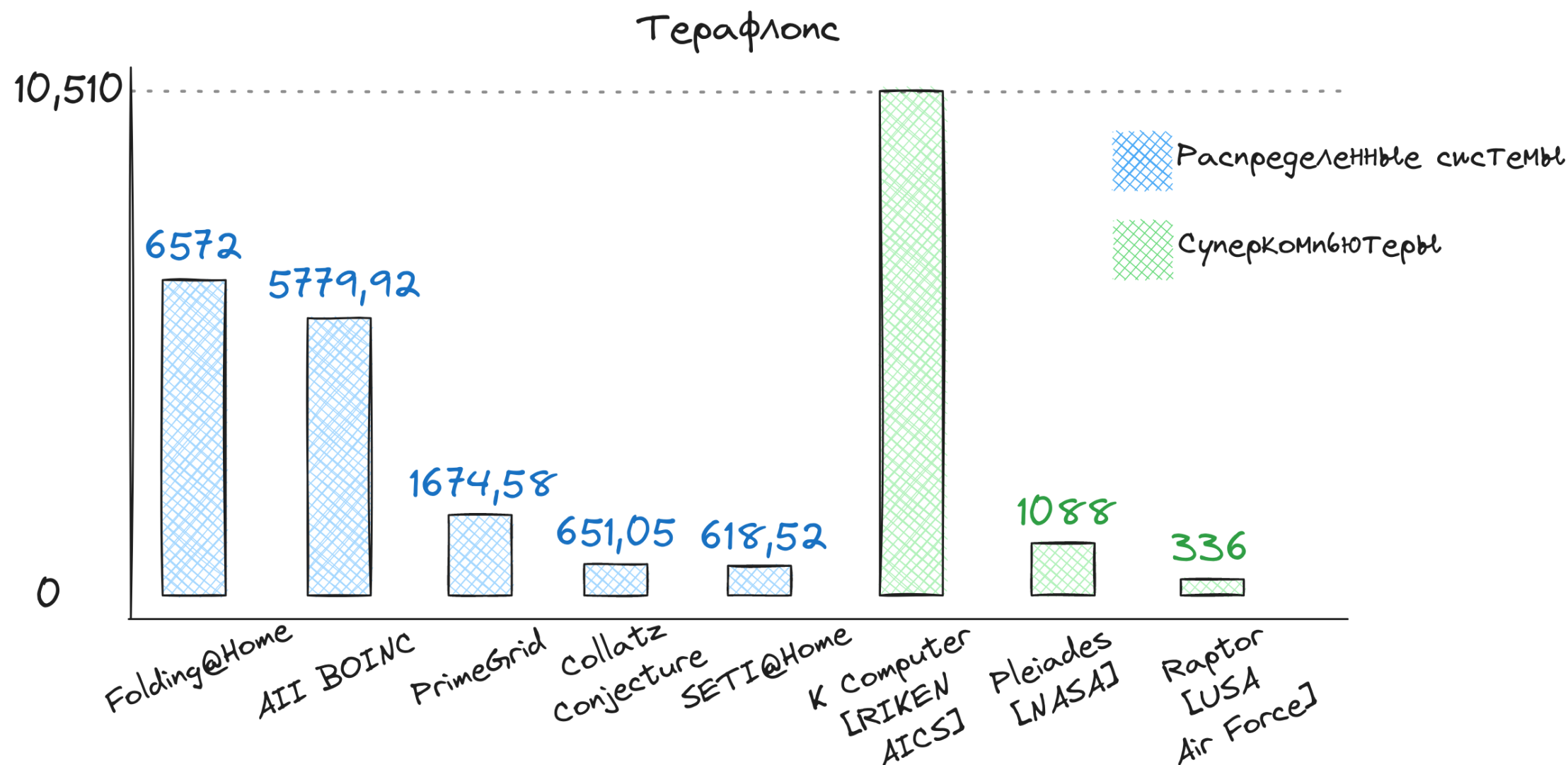
- Провести обзор существующих систем распределенных вычислений;
- Провести анализ подходов к проектированию распределенных вычислений;
- Сформулировать критерии сравнения методов распределенных вычислений;

Особенности распределенных систем

Особенности распределенных систем:

- Распределенная система — это вычислительная среда, в которой многочисленные компоненты расположены на нескольких вычислительных устройствах в сети.
- Мотивацией роста распределенных вычислений является доступность недорогих, высокопроизводительных компьютеров и сетевых инструментов.
- Распределенная система может обладать более высокой производительностью, чем один конкретный суперкомпьютер
- Основным компонентом во всех архитектурах распределенных вычислений является понятие связи между узлами системы.

Сравнение производительности различных вычислительных систем.



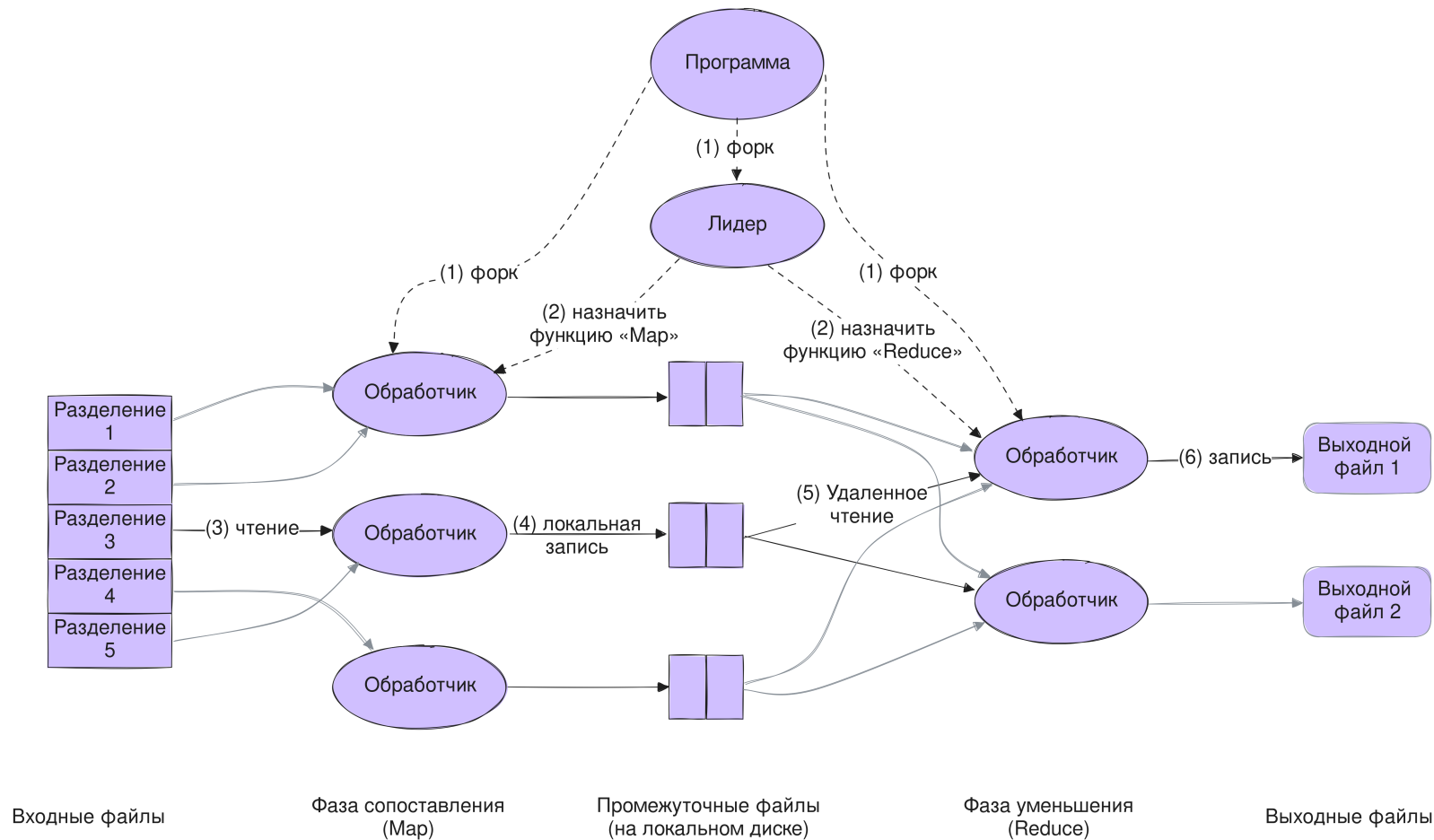
Существующие решения: Hadoop MapReduce

Программная модель

- Вычисления в MapReduce используют набор входных пар ключ/значение и создает набор выходных пар ключ/значение.
- Функция Map принимает входную пару и создает набор промежуточных пар ключ/значение
- Функция Reduce принимает промежуточный ключ k и набор значений для этого ключа, после чего объединяет эти значения для формирования возможно меньшего набора значений

Существующие решения: Hadoop MapReduce

Реализация



Существующие решения: Spark

Программная модель

- Основной абстракцией в Spark является устойчивый распределенный набор данных (Resilient Distributed Dataset – RDD).
- RDD — это доступная только для чтения секционированная коллекция записей. RDD могут быть созданы только с помощью детерминированных операций либо с данными в некотором стабильном хранилище, либо с другими RDD.
- Вторая важная абстракция – параллельные операции с этими наборами данных, вызываемые путем передачи функции для применения к набору данных.
- Операции разделяются на два типа: преобразования – отложенные (ленивые) операции, и действия, которые запускают вычисление для возврата значения программе или записи данных во внешнее хранилище

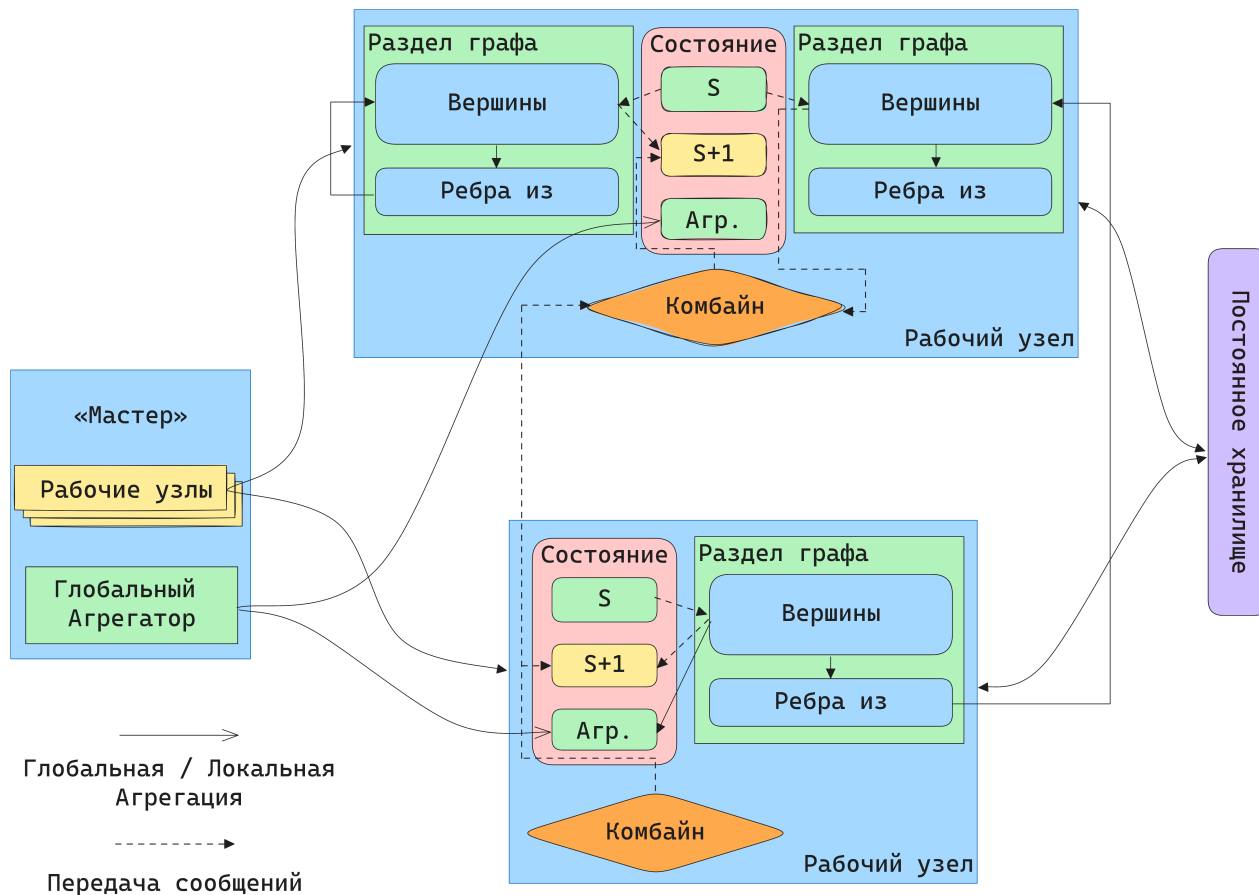
Существующие решения: Pregel

Программная модель

- Входными данными для вычислений в Pregel является ориентированный граф, в котором каждая вершина однозначно идентифицируется строковым идентификатором вершины.
- Вычисления в Pregel состоят из последовательности итераций, называемых супершагами, разделенных точками глобальной синхронизации. Во время супершага фреймворк вызывает определенную пользователем функцию для каждой вершины, практически параллельно.
- Завершение алгоритма основано на том, что каждая вершина графа голосует за остановку. Вершина деактивирует себя, проголосовав за остановку алгоритма. Алгоритм завершается, когда все вершины одновременно неактивны и нет передаваемых сообщений.

Существующие решения: Pregel

Pregel делит граф на разделы, каждый из которых состоит из набора вершин и всех исходящих ребер этих вершин. На каждом шаге каждая вершина считывает все входящие сообщения и устанавливает максимальное значение для своего текущего значения. Затем он отправляет это максимальное значение по всем своим ребрам. Если значение в узле не изменяется во время супершага, узел голосует за остановку.



Существующие решения: DryadLINQ

Программная модель

- Термин LINQ относится к набору конструкций .NET для манипулирования наборами и последовательностями элементов данных.
- Dryad в свою очередь выступает как механизм распределенного выполнения LINQ выражений.
- DryadLINQ сохраняет модель программирования LINQ и расширяет ее для параллельного программирования, определяя небольшой набор новых операторов и типов данных.
- Все функции, вызываемые в выражениях DryadLINQ, должны быть свободны от побочных эффектов.

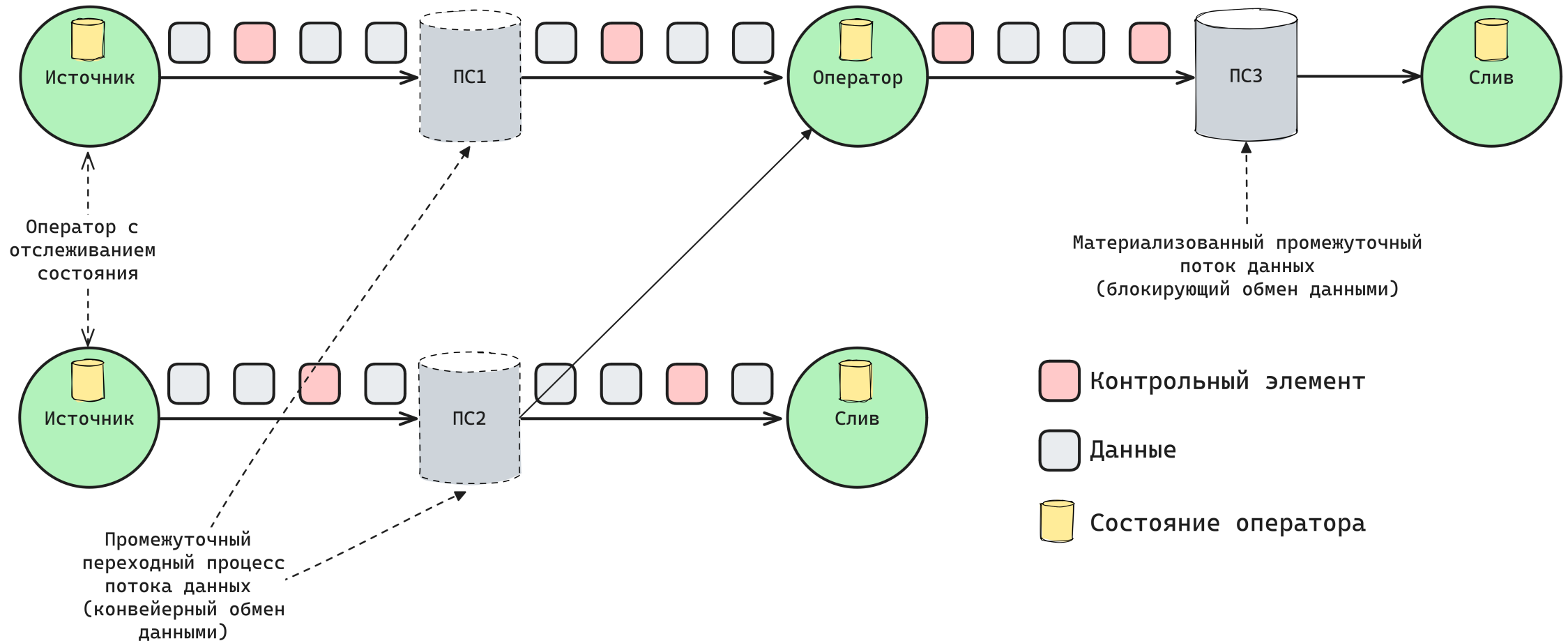
Существующие решения: Google Dataflow

Программная модель

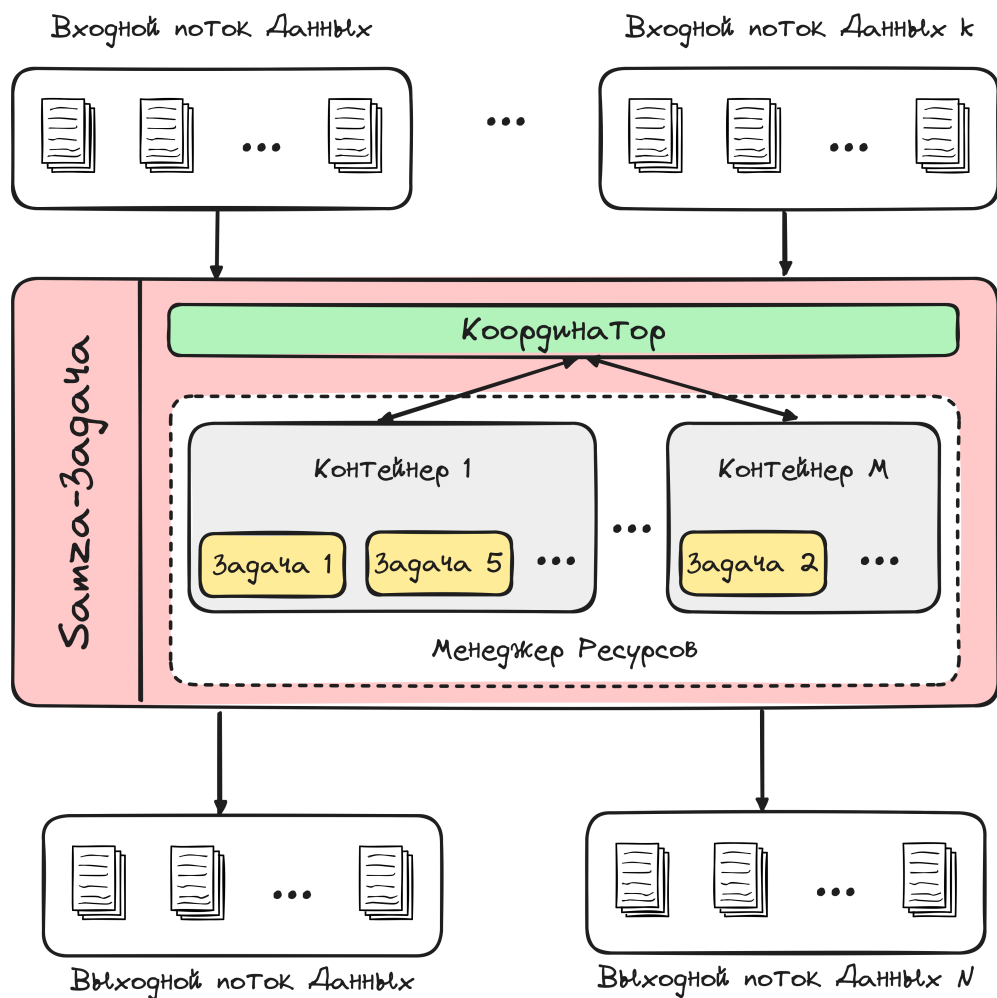
- Модель Dataflow содержит две основных операций, которые работают с парами ключ-значение, проходящими через систему: ParDo для универсальной параллельной обработки и GroupByKey для группируемых пар ключ-значение по ключу.
- Операция ParDo работает поэлементно с каждым входным элементом и, таким образом преобразуется в работу с потоковыми данными.
- Операция GroupByKey, с другой стороны, собирает все данные для данного ключа перед отправкой их вниз по потоку.
- В случае если источник входных данных неограничен, то используется обработка данных окнами.

Существующие решения: Apache Flink

Программная модель



Существующие решения: Apache Samza



Задача в Samza состоит из набора экземпляров виртуальной машины Java, каждый из которых обрабатывает подмножество входных данных. Samza представляет задачи в виде ориентированного графа операторов (вершин), соединенных потоками данных (ребрами). Оператор — это преобразование одного или многих потоков в другой поток (потоки).

Классификация методов распределенных вычислений

Классификация моделей по сценариям использования

Модель	Сценарий использования
MapReduce	Пакетная обработка больших наборов данных
Apache Spark	Пакетная обработка данных в реальном времени и итеративные алгоритмы
Pregel	Обработка графов большого масштаба
DryadLinq	Распределенная обработка данных общего назначения, научные вычисления
Google Dataflow	Непрерывная обработка данных в реальном времени, приложения, управляемые событиями
Apache Flink	Обработка потоков данных, приложения, управляемые событиями, аналитика в реальном времени
Apache Samza	Обработка потоков данных, приложения, управляемые сообщениями

Выводы

В результате данной работы:

- был проведен обзор существующих систем распределенных вычислений;
- был проведен анализ подходов к проектированию распределенных вычислений;
- были сформулированы критерии сравнения методов распределенных вычислений.

Цель — классифицировать существующие методы распределенных вычислений — была выполнена.