APPROACHES TO USING THE DASK AND NUMPY LIBRARIES TO PERFORM OPERATIONS ON NORMALLY DISTRIBUTED ARRAYS.

Есикова Т.Н. (ИЭОПП СО РАН), Yuldashov Sh.N. (TUIT), Ismailov Sh.R. (TUIT)

Annotation. This article explores the principles of parallel programming in Python using Dask. Parallel processing means running multiple tasks at the same time using multiple processors on the same computer.

Key words: python, dask, parallel computing, normally distributed, divide into parts, massive, parallel programming, effectiveness.

ПОДХОДЫ К ИСПОЛЬЗОВАНИЮ БИБЛИОТЕК DASK И NUMPY ДЛЯ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ НАД НОРМАЛЬНО РАСПРЕДЕЛЕННЫМИ МАССИВАМИ.

Есикова Т.Н. (ИЭОПП СО РАН), Юлдашов Ш.Н. (ТУИТ), Исмаилов Ш. (ТУИТ)

Аннотация. В этой статье изучаются принципы параллельного программирования на Python с использованием Dask. Параллельная обработка означает одновременное выполнение нескольких задач с использованием нескольких процессоров на одном компьютере.

Ключевые слова: python, dask, параллельные вычисления, нормально распределенные, разбиение на части, массивность, параллельное программирование, эффективность.

Введение. Dask - это универсальный инструмент, который поддерживает различные рабочие нагрузки. Он состоит из двух частей: Динамическое планирование задач, оптимизированное для вычислений. Это похоже на Airflow, Luigi, Celery или Make, но оптимизировано для интерактивных вычислительных рабочих нагрузок. Большие коллекции данных, такие как параллельные массивы, фреймы данных и списки, которые расширяют общие

интерфейсы, такие как NumPy, Pandas или итераторы Python, до больших объемов памяти или распределенных сред. Эти параллельные коллекции выполняются поверх динамических планировщиков задач. Гибкая библиотека для параллельных вычислений на Python.

Numpy. Помимо очевидного научного применения, NumPy также может быть использован в качестве эффективного многомерного контейнера общих данных. Могут быть определены произвольные типы данных. Это позволяет NumPy легко и быстро интегрироваться с широким спектром баз данных. фундаментальный пакет для научных вычислений с использованием Python.

NumPy и Dask: в чем разница?

NumPy и Dask можно отнести к категории инструментов "Data Science". Вот некоторые из функций, предлагаемых NumPy:
мощный объект N-мерного массива, сложные (широковещательные) функции, инструменты для интеграции кода C/C++ и Fortran, С другой стороны, Dask предоставляет следующие ключевые функции:
Поддерживает различные рабочие нагрузки, Динамическое планирование задач, Тривиально настроить и запустить на ноутбуке за один процесс.

Spark - это быстрый и универсальный механизм обработки, совместимый с данными Hadoop. Он может работать в кластерах Hadoop через автономный режим YARN или Spark и может обрабатывать данные в HDFS, HBase, Cassandra, Hive и любом формате ввода Hadoop. Он предназначен для выполнения как пакетной обработки (аналогично MapReduce), так и новых рабочих нагрузок, таких как потоковая передача, интерактивные запросы и машинное обучение.

Гибкая и мощная библиотека анализа и обработки данных для Python, предоставляющая помеченные структуры данных, аналогичные объектам R data.frame, статистические функции и многое другое.

Это совместная работа Apache Spark и Python. это Python API для Spark, который позволяет вам использовать простоту Python и мощь Apache Spark для управления большими данными.

Celery - это асинхронная очередь задач / очередь заданий, основанная на распределенной передаче сообщений. Он ориентирован на работу в режиме реального времени, но также поддерживает планирование.

Используйте Airflow для создания рабочих процессов в виде ориентированных ациклических графиков (DAG) задач. Планировщик воздушного потока выполняет ваши задачи для массива рабочих, следуя указанным зависимостям. Расширенные утилиты командной строки упрощают выполнение сложных операций с DAG. Богатый пользовательский интерфейс позволяет легко визуализировать конвейеры, работающие в процессе производства, отслеживать ход выполнения и при необходимости устранять неполадки.

Что еще более важно, мы пытаемся дать представление о том, как работают эти различные методы и когда их следует использовать.

1) Numpy быстрее, чем Dask для меньшего количества элементов;

2) Dask берет на себя Numpy примерно на 1e7 элементов;

3) Numpy не может выдавать результаты для большего количества элементов, так как не может поместить их в память.

Но по мере того, как ваши данные становятся больше, чем то, что вы можете поместить в ОЗУ, pandas будет недостаточно.

Dask — это библиотека с открытым исходным кодом, которая обеспечивает расширенное распараллеливание для аналитики, особенно при работе с большими данными.

Параллельная обработка означает одновременное выполнение нескольких задач с использованием нескольких процессоров на одном компьютере.

Мы посмотрим эффективность решения параллельного вычисления с помощью dask в следующем примере.

Пример. Создайте массив 20000x20000 нормально распределенных случайных значений, разбитых на куски размером 1000x1000;

Возьмите среднее значение по одной оси;

Возьмите каждый 100-й элемент.

Dask.array — это библиотека, похожая на NumPy, которая выполняет такие трюки для работы с большими наборами данных, которые не помещаются в память.

Dask.array переводит ваши операции с массивами в граф взаимосвязанных задач с зависимостями данных между ними. Затем Dask выполняет этот граф параллельно с несколькими потоками.

```
import dask.array as da
import numpy as np
x = da.random.normal(10, 0.1, size=(20000, 20000), chunks=(1000, 1000) )
# Массив из 400 миллионов элементов
# Разрезать на куски размером 1000x1000
y = x.mean(axis=0)[::100]  # Выполнение операций в стиле NumPy
x.nbytes / 1e9  # Гигабайты ввода обработаны лениво
```

Сравнение производительности. Следующий эксперимент был проведен на тяжелом персональном ноутбуке. Ваша производительность может отличаться. Если вы пытаетесь использовать версию NumPy, убедитесь, что у вас более 4 ГБ основной памяти.

```
import numpy as np
%%time
x = np.random.normal(10, 0.1, size=(20000, 20000))
y = x.mean(axis=0)[::100]
```

y

CPU times: user 33 s, sys: 4.98 s, total: 38 s

Wall time: 38.3 s

NumPy: 38.3 с, требуется гигабайт памяти

```
import numpy as np
```

```
%%time
x = np.random.normal(10, 0.1, size=(20000, 20000))
y = x.mean(axis=0)[::100]
y
```

```
CPU times: user 33 s, sys: 4.98 s, total: 38 s
Wall time: 38.3 s
```

import dask.array as da

%%time

x = da.random.normal(10, 0.1, size=(20000, 20000), chunks=(25, 25))

y = x.mean(axis=0)[::100]

y.compute()

CPU times: user 36.5 s, sys: 147 ms, total: 36.6 s

Wall time: 19.1 s

Dask Array: 19.1 с, требуется мегабайт памяти

```
import dask.array as da
```

```
%%time
x = da.random.normal(10, 0.1, size=(20000, 20000), chunks=(1000, 1000))
y = x.mean(axis=0)[::100]
y.compute()
```

```
CPU times: user 36.5 s, sys: 147 ms, total: 36.6 s
Wall time: 19.1 s
```

Заключение. Обратите внимание, что вычисление массива Dask выполнялось за 19.1 секунды, но использовало 36.5 секунды процессорного времени пользователя. Вычисление numpy выполнялось за 38.3 секунды и использовало 33 секунды процессорного времени пользователя.

Dask закончил быстрее, но использовал больше общего процессорного времени, потому что Dask смог прозрачно распараллелить вычисления из-за размера фрагмента.

1. Data Science with Python and Dask 1st Edition by Jesse Daniel.
2. Роберт Юханссон. Numerical Python: Scientific Computing and Data Science.
3. Воеводин В.В. Численные методы, параллельные вычисления и информационные технологии. М.: БИНОМ, 2008.
4. С.В.Андреев [и др.] Реализация параллельного алгоритма решения задач оптимизационного анализа // Препринты ИПМ им. М.В.Келдыша. 2014. № 101. 11 с. URL: http://library.keldysh.ru/preprint.asp?id=2014-101.
5. Никифоров А.И., Садовников Р.В. Параллельные вычисления на гибридной вычислительной системе в задачах двухфазной фильтрации // Выч. мет. программирование, 2018, том 19, выпуск 1, 9–16.
6. А. В. Старченко, В. Н. Берцун, Методы параллельных вычислений, Издательство Томского университета. 2013. — 223 с.
7. Гергель В.П. Теория и практика параллельных вычислений. М.: БИНОМ, 2007.

ПОДХОДЫ К ИСПОЛЬЗОВАНИЮ БИБЛИОТЕК DASK И NUMPY ДЛЯ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ НАД НОРМАЛЬНО РАСПРЕДЕЛЕННЫМИ МАССИВАМИ.

Есикова Т.Н. (ИЭОПП СО РАН), Юлдашов Ш.Н. (ТУИТ), Исмаилов Ш. (ТУИТ)

APPROACHES TO USING THE DASK AND NUMPY LIBRARIES TO PERFORM OPERATIONS ON NORMALLY DISTRIBUTED ARRAYS.

Есикова Т.Н. (ИЭОПП СО РАН), Yuldashov Sh.N. (TUIT), Ismailov Sh.R. (TUIT)

Муаллифлар ҳақида маълумот

Татьяна Николаевна Есикова
Институт экономики и организации промышленного производства СО РАН, 630090, Россия, г. Новосибирск, проспект Академика Лаврентьева, 17, кандидат экономических наук, ведущий научный сотрудник, тел. (383)333-25-96, e-mail: T.N.Yesikova@gmail.com

Юлдашов Шавкат Нуриддин угли - ассистент кафедры Информационных технологий.
Номер телефона: +998910771328.
Email: shavkatbek28.12@gmail.com

Исмаилов Шихназар Рашид угли - ассистент кафедры Мультимедийных технологий.
Номер телефона: +998974532210.
Email: tuit.9393@gmail.com