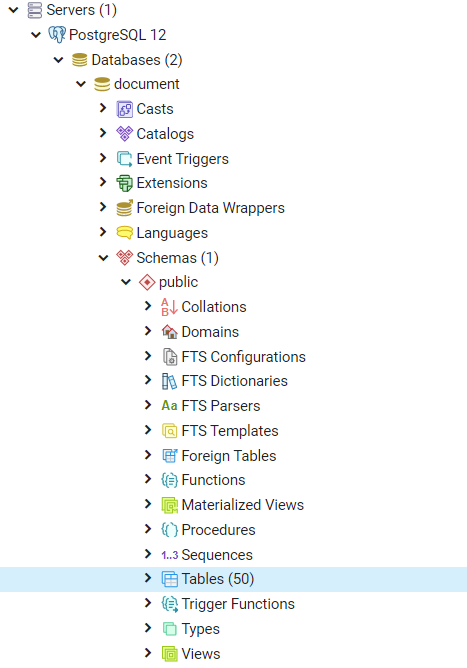
В работе представляются результаты экспериментов оценки быстродействия параллельных реализаций парсинга и записи полученных данных в базу данных, в среде бюджетного персонального компьютера (Приложение – Сводная таблица характеристик, тестируемых ЭВМ).

Было получено 2 исходных файла (учебный план), прошлых лет, с которыми и производилась оценка затрат времени и составление таблиц и парсинг данных из plx файлов в бд.

Из данных файлов были выявлены соответствующие поля и по ним были созданы таблицы в бд с соответствующими признаками.

Были выявлены следующие таблицы: ПланыНовыеЧасы, Уровень\_образования, псСтандарты, СправочникВидыПрактик, Компетенции, псГруппы, псТипыХарактеристикФункции, псОбобщенныеФункции, Заезды, ПланыРазбиения, Факультеты, ДолжностныеЛица, ПланыКомпетенцииДисциплины, СправочникТипаЧасов, Документ, ПланыПрофСтандарты, ПланыГрафикиЯчейки, diffgram, ПараметрыПлана, псФункции, СкрытьВидыРабот, КратностьЧасов, Филиалы, ПланыЦиклы, before, СправочникТипБлока, ДолжЛица\_Планы, СправочникБазы, ПланыНовыеГрафики, ПланыКомпетенции, ПланыКомпетенцииПрофСтандарты, ЗаездыНормы, СправочникВидОбъекта, ПланыКонтингент, СправочникТипаРабот, ПрограммаПодготовки, ФормаОбучения, псХарактеристикиФункций, ВидыДеятельности, ПланыСтроки, Кафедры, ПланыПрофили, ВыбранныеВидыРабот, ООП, УровеньОбразования, СправочникВидыРабот, СправочникТипОбъекта, ПланыВидыДеятельности, Планы.



В задаче экспериментирования с процедурами параллельного вычисления имеется два критериальных параметра: время вычисления и степень ускорения, определяемая отношением времени последовательной обработки ко времени параллельной обработки.

Значения критериальных параметров зависят от двух факторов: размера рабочей нагрузки на процедуру и количества параллельных ветвей.

Для эксперимента, был написан парсер, на языке Python, xml(plx) файлов, который записывает данные из таблиц (тэгов) в бд.

Так как текущий data set из 2-ух файлов маловат, был написан маленький скрипт по генерации копий файлов с заменой названия (лежит в папке genfiles).

Стоит учесть, то что файлы разные по размер и это будет сказывается на скорости обработки, поэтому было принято решения проверить сначала на более простых файлах, после чего на более тяжеловесных файлах. Ниже будут приведены таблицы результатов по 2-ум типам файлов.

Первый тест будет произведен с файлом 09.03.01.02-1234-2016.plx с размером 829КБ и до 150 копий фала.

Таблица 1. Результаты замеров времени (c) вычисления при варьировании числа потоков,

числа ядер и гранулярности задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Потоки** | **10** | **25** | **50** | **100** | **150** |
| **1** | 7,2677302 | 18,5514128 | 36,7133557 | 82,0052709 | 110,6746280 |
| **2** | 5,5745096 | 11,8321518 | 27,2187612 | 43,9257984 | 65,4179015 |
| **3** | 4,2266352 | 9,4364001 | 18,4481730 | 35,6156911 | 54,6968827 |
| **4** | 3,6308805 | 9,6784853 | 15,7115521 | 31,4356100 | 44,7100415 |

Рис. 1. Графики коэффициентов ускорения при варьировании числа потоков

Второй тест будет произведен с файлом 09.03.01\_02-2019.plx с размером 3082КБ и до 150 копий фала.

Таблица 2. Результаты замеров времени (c) вычисления при варьировании числа потоков,

числа ядер и гранулярности задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Потоки** | **10** | **25** | **50** | **100** | **150** |
| **1** | 30,3956904 | 76,0451130 | 152,6322774 | 312,3548293 | 449,4782640 |
| **2** | 17,9967038 | 43,4448020 | 85,6357100 | 181,7410304 | 272,0168783 |
| **3** | 16,0176477 | 38,7935459 | 80,5575785 | 152,7809765 | 235,8286857 |
| **4** | 13,2225451 | 35,3233423 | 62,3538620 | 131,6907467 | 198,1465356 |

Рис. 2. Графики коэффициентов ускорения при варьировании числа потоков

Для параллелизма на Python была использована библиотека multiprocessing, для замера времени time (Функция time() модуля time вернет время в секундах с начала эпохи как число с плавающей запятой). В multiprocessing процессы порождаются путём создания объекта Process и последующего вызова его метода start().

Кратко о директории проекта Upload:

1. Папка Backup, в ней 2 файла, резервная копия бд – document.sql и sql скрипты по созданию таблиц – sql
2. Папка files, в которой хранятся файлы на загрузку в бд
3. Папка genfiles, в ней скрипт генерирует файлы для нагрузки, по 2 исходным файлам (09.03.01\_02-2019.plx и 09.03.01.02-1234-2016.plx)
4. Папка гр, в ней таблицы и графики результатов после эксперимента
5. changeEncoding.py, файл по изменению кодировки внутри файла, для дальнейшего парсинга
6. conf.ini, переменны бд
7. upload.py, файл по парсингу и записи в бд

Есть моменты по улучшению. К примеру:

1. Сделать декларативную проверку схемы бд и содержащихся таблиц в файле на наличия присутствия

Самый первый вариант был вовремя парсинга проверять наличие таблиц и столбцов таблицы в бд, но это ведёт к сильной нагрузке бд из-за множества запросов к бд. Поэтому продуктивнее заранее получать схему файла и сверять ее с бд, а после записывать данные.

1. Делать 1 запрос на 1 файл. Подготовить все данные по файлу и отправить одним запросом

upload.py работает сейчас по принципу, если есть файлы на загрузку которых нету в бд грузим их, после чего происходит разбивка списка файлов на ядра компьютера. Там уже по получившемуся списку происходит парсинг, выявления таблиц, и сбор всех признаков данной таблицы после чего происходит запись в бд (т.е. загрузку производится по 1 таблице из файла)

1. Модернизировать модель базы данных

**Приложение**

Сводная таблица характеристик, тестируемых ЭВМ

|  |  |
| --- | --- |
| Наименование характеристик | Компьютер 1 |
| Тип процессора | DualCore Intel Core i5-7200U |
| Исходная частота | 2500 МГц |
| Максимальная частота | 3100 MHz |
| Архитектура процессора | Kaby Lake-U |
| Cash-памяти | Кэш L1 — 2x32+2x32 Кб per core  Кэш L2 — 256x2 Кб per core (On-Die, ECC, Full-Speed)  Кэш L3 — 3 Мб (On-Die, ECC, Full-Speed) |
| Тип сокета | BGA1356 |
| Напряжение питания процессора | 0.6 V |
| Мощность процессора | 15 W |
| Количество ядер/потоков | 2/4 |
| Производитель | Asus |
| Материнская плата | Asus VivoBook X556UQK |
| Наличие и тип слотов PCI | PCI  PCI Express 2.0 |
| Тип видеокарты (встроенная, дополнительная) | Intel HD Graphics 620  NVIDIA GeForce 940MX |
| Тип шины | Встроено  PCI-устройство 8086-5916 / 1043-1490 (Rev 02) |
| Объем памяти видеокарты | 1 Гб  2 Гб |
| Физическая память ЭВМ | HGST HTS721010A9E630 (1 ТБ, 7200 RPM, SATA-III) |
| Оперативная память (объем, тип и скорость) | 8Гб SO-DIMM DDR4 15-15-15-35 4-50-17-8 2T  DDR3 SO-DIMM DDR4  2x 1ГГц (2.13ГГц)  Crucial DDR4 16G 2666 |
| Драйвер видеокарты | 23.20.16.4905 |
| Интерфейсы ввода-вывода | USB 2.0, USB 3.0, USB type C, HDMI, microSD, Audio 3.5, LAN, VGA |
| ОС | Windows 10 |