HADOOP MAPREDUCE ORQALI KATTA HAJMLI MA'LUMOTNI PARALLEL QAYTA ISHLASH

Xudayshukur Shavkat o'g'li Quzibayev
Muhammad Al Xorazmiy nomidagi Toshkent Axborot Texnologiyalari Universiteti
xudayshukur66@gmail.com

Tohir Quronbayevich O'razmatov
Muhammad Al Xorazmiy nomidagi Toshkent Axborot Texnologiyalari Universiteti
Urganch filiali
tohir20314@gmail.com

Bonuraxon Baxromovna Nurmetova
Muhammad Al Xorazmiy nomidagi Toshkent Axborot Texnologiyalari Universiteti
Urganch filiali
bonuraxon20102018@gmail.com

ANNOTATSIYA

Ushbu maqolada biz katta hajmli ma'lumot sifatida qarash mumkin bo'lgan tarixiy asardagi so'zlarning chastotaviy tahlilini amalga oshirdik. Buning uchun katta hajmdagi ma'lumotlarni taqsimlangan saqlash tizimlari saqlash jarayonini amalga oshirdik, hamda parallel hisoblashlar yordamida qayta ishlangan ma'lumotlani tahlil qildik. Taqsimlangan saqlash tizimi sifatida Hadoop HDFS(Hadoop Distributed File System) tizimidan, parellel hisoblashni amalga oshirishda esa Hadoop MapReduce komponentidan foydalanilgan. Bundan tashqari ushbu maqolada, aynan shu katta hajmli ma'lumotlarni ananaviy hisoblashlar yordamida qayta ishlashdan olingan natijalar ham keltirilgan. Ananviy hisoblashlar va parallel hisoblashlar yordamida olingan natijalarga asoslangan holda xulosalar qilingan.

Kalit so'zlar: so'zlar chastotasi, Big Data, Hadoop HDFS, Hadoop MapReduce, parallel hisoblash, taqsimlangan saqlash tizimi

ABSTRACT

The difficulty of processing semi-ordered massive quantities of data with distributed storage systems and parallel computing is addressed in this article. Hadoop HDFS (Hadoop Distributed File System) is used as a distributed storage system, while Hadoop MapReduce is utilized for parallel processing. Furthermore, the outcomes of processing these massive amounts of data using non-parallel algorithms are provided in this study. The gathered results were used to draw conclusions.

Keywords: Hadoop HDFS, Hadoop MapReduce, Big Data, parallel computing, distributed storage system.

KIRISH

Dunyoda raqamlangan ma'lumotlar hajmi shiddat bilan o'sib bormoqda. Bu o'z navbatida raqamli ma'lumotlarni saqlab qo'yich, ularni saralash, qayta ishlash va ular asosida xulosalar chiqarish kabi muommolarni yuzaga chiqaradi. Bu muommolarni o'rganish va yechimlar taklif qilish uchun axborot texnologiyalari sohasida Big data, Data science (malumotlar ilmi), Data mining (ma'lumotlarni intelektual tahlili), Machine learning (mashinali o'qitish), Deep learning (chuqur o'qitish), Sun'iy neyron tarmog'i kabi fan tarmoqlari vujudga keldi. Biz ushbu maqolada tadqiq qilgan qilgan muommo BigData (katta hajmli ma'lumotlar) sohasiga tegishli. Hozirgi kundagi ma'lumotlarning keskin oshib borish fonida, ularni saqlash va tezkor qayta ishlash masalasi mavuning dolzarbligini ko'rsatadi.

Tadqiqot obyekti sifatida o'zbek adibi Abdulla Qodiriyning "O'tkan kunlar" asarini katta hajmli ma'lumot sifatida belgilab oldik. Tadqiqot predmeti sifatida esa katta hajmli ma'lumotlarni saqlash uchun ishlatiladigan Apache Hadoop HDFS hamda ma'lumotlarni parallel qayta ishlovchi Hadoop MapReduce dasturlarini belgilab oldik. Izlanishlarimizning maqsadi sifatida katta hajmli ma'lumotlarni ananaviy hisoblash usullari orqali qayta ishlab bo'lmasligini, parallel hisoblashlar orqali qayta ishlash samarali va tezkor ekanligini isbotlash.

Izlanishlarimizning vazifalari sifatida quyidagilarni belgilab oldik:

- Katta hajmli ma'lumotni taqsimlangan fayl tizimlarida saqlash
- Katta hajmli ma'lumotni ananaviy usulda qayta ishlab natija olish
- Katta hajmli ma'lumotni parallel hisoblash yordamida qayta ishlab natija olish
- Olingan natijalarni solishtirib xulosalar chiqarish

Obyekt sifatida belgilab olganimiz Abdulla Qodiriyning "O'tkan kunlar" asarining elektron shakldagi talqinini topamiz. 220 betdan iborat elektron matn shakldagi asarni .txt formatiga o'tkazib olamiz. Matndagi so'zlarni chastotasini ya'ni har bir so'zning takrorlanishlar sonini aniqlovchi dasturni Java dasturlash tilida yozib olamiz. Tanlangan obyektimizni ananaviy usulda qayta ishlaymiz. Olingan natijani va qayta ishlash vaqtini qayt qilib qo'yamiz. Endi ayni shu katta hajmli ma'lumotimizni Hadoop MapReduce modeli yordamida parallel qayta ishlaymiz. Olingan natijalarni va qayta ishlash uchun sarflangan vaqtni yana qayt qilib qo'yamiz. Qayd qilingan natijalarni va qayta ishlar uchun sarflangan vaqtlarni solishtirib ko'rganimizda yaqqol farqni kuzatdik. Olingan natijalarni solishtirish natijasida tegishli xulosalar qildik.

ADABIYOTLAR TAHLILI VA METODOLOGIYA

Bugungi kunda har kuni 2,5 (1018) kvintillion bayt ma’lumot yaratilmoqda va bu ko’rsatgich 2022 yilda har bir inson uchun kuniga 2,1 MB ma’lumot yaratilganligini bildiradi.[1] Bu turdagi katta hajmli ma’lumotlar bilan ishlashda yangidan-yangi algoritm va texnologiyalar ishlab chiqishni talab qilmoqda. 2018 yilda jami to‘plangan ma’lumot miqdori 912 eksabaytni tashkil etdi, deb xabar beradi TrendFocus[2]. 2013-2015 yillarda oralig‘ida to‘plangan ma’lumotlar hajmi shundan oldingi butun insoniyatning o‘tmish tarixiga qaraganda ko‘proq ma’lumotlar yig‘ilganini takidlashgan. 2025 yilga kelib, barcha ma’lumotlar 163 zettabayt (ZB) ga teng bo‘lishi mumkinligi ta’kidlangan.

Katta ma’lumotlar - bu xar xil turdagi va avtonom ma’lumot manbalaridan kelib chiqadigan keng miqyosli, hajmli va ko‘p formatli ma’lumot oqimlarining yig‘indisidir[2,3]. Katta hajmli ma’lumotlarning asosiy xarakteristikasi bo‘lib, u keng miqyosli ma’lumotlar markazlarida va saqlash zonalari tarmoqlarida saqlash joylarini egallash bilan tavsiflanadi. Katta ma’lumotlarning katta o‘lchamlari nafaqat ma’lumotlarning turli xil bo‘lishiga olib keladi, balki natijada ma’lumotlar to‘plamida xilma-xil o‘lchovlar paydo bo‘ladi[4]. Katta miqdordagi ma’lumotlarni tahlil qilish inson his etish imkoniyatidan tashqarida bo‘lgan qonuniyatlarni aniqlashda yordam beradi[5]. Big data atamasi ilk bora Nature jurnalining 2008 yildagi sonida duch kelish mumkin. Jurnal muharriri Klifford Linch dunyodagi ma’lumotlar hajmining intensiv ortib borishiga bag’ishlangan maqolasida bu haqda to’xtalgan. Mutaxassislarning fikricha, kuniga 100 gb dan ko'p ma'lumot tushadigan oqimlarga big data deb aytish mumkin. Katta hajmli ma’lumotlarni tushintirishda “Meta Group”(eski Facebook) kompniyasi tomonidan ishlab chiqilgan xususiyatlar muhimdir.

1-rasm. Katta hajmli ma’lumot xususiyatlari.

- Volume – ma’lumotlar hajmining kattaligi [3]. Ma’lumotlarning hajmini kattaligi, ahamiyati va uni katta ma’lumotlar deb hisoblash mumkinmi yoki yo‘qligini birdiradi;

- Variety – bu ma’lumotlarning turi va xususiyatini ifodalab, turli xil ma’lumotlarni bir vaqtning o‘zida qayta ishlash imkoniyatidir.
- Velocity – ma’lumotlar o‘sish tezligi va natijaga erishish uchun ma’lumotlarni qayta ishlash vaqtining real vaqtga yaqinligi.
- Value – Katta ma’lumotlar to‘plamlarini qayta ishlash va tahlil qilish orqali erishish mumkin bo‘lgan ma’lumotlarning ahamiyati.
- Veracity – bu katta ma’lumotlar uchun kengaytirilgan ta’rif bo‘lib, bu ma’lumotlar sifati va ma’lumotlar qiymatini anglatadi.

Ushbu xususiyatlardan kelib chiqib biz tanlagan obyekt Abdulla Qodiriyning “O’tkan kunlar” asarini katta hajmli ma’lumot deb atash mumkin. Ushbu asar 220 sahifadan iborat bo’lib, unda sal kam 100 000 so’zdan foydalanilgan. Belgilar soni esa 574 000 dan oshadi. Biz yechmoqchi bo’lgan masala esa ushbu asardagi so’zlarning chastotasini hisoblashdan iborat. Boshqacha aytganda, ushbu ulkan asarda har bir so’z nechta marta qo’llanganini xiosblash zarur bo’ladi.

Bu masalani yechishda biz ikki hil metoddan foydalandik:

1. Java Core ga asoslangan dastur yordamida ananviy hisoblash
2. Hadoop MapReduce ga asoslangan parallel hisoblash

Endi bu ikki metod haqida to’xtalib o’tamiz. Java Core ga asoslangan dasturimiz Eclipse IDE muhitida yozilgan. Bu dastur bitta WordCount deb nomlangan klass dan tuzilgan bo’lib, java.io.FileInputStream kutubxonasi yordamida katta hajmli ma’lumotni fayldan o’qib oladi. Bundan tashqari dasturda java.util.ArrayList, java.util.Iterator, java.util.Scanner kabi kutubxonalardan foydalanilgan. Dasturning asosiy bajaruvchi tanasi quyidagicha

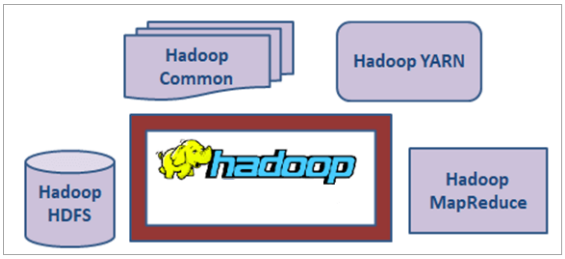
```
while (fileinput.hasNext()) {
    String nextword= fileinput.next();

    if(words.contains(nextword)) {
        int index=words.indexOf(nextword);
        count.set(index, count.get(index)+1);
    }
    else {
        words.add(nextword);
        count.add(1);
    }
```

Dastur sanalgan so’zlarni java.io.FileOutputStream kutubxonasi yordamida faylga yozib qo’yadi. Dastur hisoblashlarni ananviy tarzda parallel bo’lmagan usulda bajaradi. Ya’ni dastur kodini kompilyatsiya qiladi. Keyingi qadamda uni JRE (Java ishlash muhiti) ga uzatadi. JRE esa o’z navbatida CPU(markaziy protsessor)ga uzatadi va CPU da hisoblash bajarilib, shu ketma ketlikda orqaga qaytadi. Java Core ga asoslangan birinchi metodimiz haqida chuqur to’xtalib o’tirmayman, ikkinchi metodimizga chuqurroq to’xtalaman.

Ikkinchi metodimiz katta hajmli ma’lumotni taqsimlangan saqlash tizimiga saqlab, uni parallel hisoblash yordamida qayta ishlashga asoslanadi. Biz buning uchun Apache litsenziyasi asosida ishlovchi Hadoop HDFS va Hadoop MapReduce dan foydalandik. Apache Hadoop - bu katta hajmdagi ma'lumotlar va hisoblash bilan bog'liq muammolarni hal qilish uchun ko'plab kompyuterlar tarmog'idan foydalanishni osonlashtiradigan ochiq manbali dasturiy ta'minot vositalari to'plami. U MapReduce dasturlash modelidan foydalangan holda katta ma'lumotlarni taqsimlangan saqlash va qayta ishlash uchun dasturiy ta'minot tizimini taqdim etadi. Apache Hadoop yadrosi Hadoop Distributed File System (HDFS) deb nomlanuvchi saqlash qismi va MapReduce dasturlash modeli bo'lgan ishlov berish qismidan iborat. Hadoop fayllarni katta bloklarga ajratadi va ularni klasterdagi tugunlar bo'ylab tarqatadi. Keyin ma'lumotlarni parallel ravishda qayta ishlash uchun paketlangan kodni tugunlarga o'tkazadi. Asosiy Apache Hadoop fremvorki quyidagi modullardan iborat:

- Hadoop Common - boshqa Hadoop modullari uchun zarur bo'lgan kutubxonalar va yordamchi dasturlarni o'z ichiga oladi;
- Hadoop Distributed File System (HDFS) - klaster bo'ylab juda yuqori agregat o'tkazish qobiliyatini ta'minlovchi tovar mashinalarida ma'lumotlarni saqlaydigan taqsimlangan fayl tizimi;
- Hadoop YARN – (2012-yilda taqdim etilgan) klasterlardagi hisoblash resurslarini boshqarish va ulardan foydalanuvchilarning ilovalarini rejalashtirishda foydalanish uchun mas’ul platforma;
- Hadoop MapReduce - keng ko'lamli ma'lumotlarni qayta ishlash uchun MapReduce dasturlash modelini amalga oshirish.
- Hadoop Ozone - (2020 yilda taqdim etilgan) Hadoop uchun ob'ektlar do'koni.



2-rasm. Hadoopning modullari

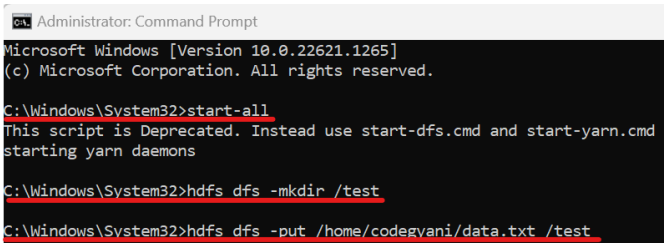
NATIJALAR VA MUHOKAMA

Hadoopning bu 4 ta modulini kompyuterda sozlab olganimizdan keyin, asardagi so’zlarni qayta ishlovchi

job(topshiriq) yaratamiz. Hadoop uchun jobni Java, Python, C++, Scala kabi dasturlash tillarida yaratish mumkin. Hadoopni kompyuterimizga sozlab olib, o'z maqsadimizga mos jobni yaratib olaganimizdan so'ng, Hadoop modullarini buyruqlar satri orqali ishga tushirib olamiz. Buning uchun buyruqlar satriga ***start-all*** buyrug'ini kiritamiz. Bu buyruqdan so'ng Hadoopning quyidagi 4 ta moduli ishga tushadi:

- Hadoop datanode
- Hadoop namenode
- Hadoop yarn nodemanager
- Hadoop yarn resourcemanager

Keyingi qadamda aynan shu buyruqlar satri yordamida HDFS da yaki jild yaratib olamiz. Buning uchun buyruqlar satriga ***hdfs dfs -mkdir /test*** buyrug'ini kiritamiz. Yangi jilda ixtiyoriy nomni berishimiz mumkin. Keyin esa katta hajmli ma'lumot sifatida belgilab olgan .txt formatidagi faylimizni HDFS da yaratgan yangi jildimizga ko'chirib o'tkazamiz. Buning uchun buyruqlar satriga ***hdfs dfs -put /home/codegyani/data.txt /test*** buyrug'ini kiritamiz.

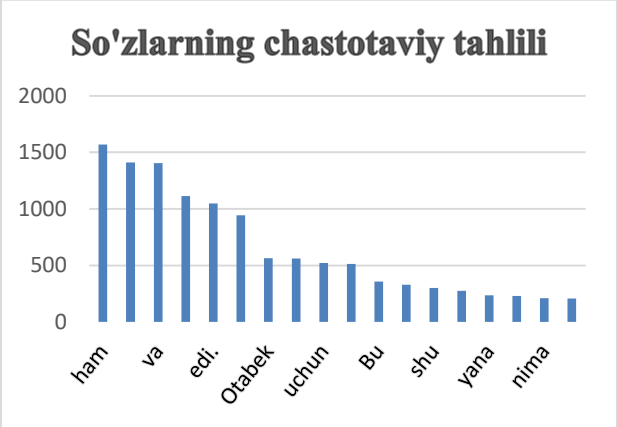


3-rasm. Hadoop modullarining ishlash jarayoni

Katta hajmli ma'lumotimizni taqsimlangan fayl tizimida saqlab olgan, endi uni qayta ishlash uchun tayyorlagan Job imizni ishga tushiramiz. Jobni buyruqlar satrida ***hadoop jar /home/codegyani/wordcountdemo.jar com.javatpoint.WC_Runner /test/data.txt /r_output***

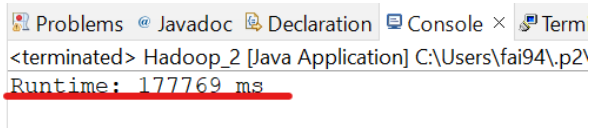
ushbu buyruq yoramida ishga tushiramiz.

O'tkazilgan tajribalar najilariga to'xtaladigan bo'lsak, natijalar absolyut bir hil chiqqanligini ko'rishimiz mumkin. Ya'ni ikkala metod bo'yicha sanalgan so'zlarning soni 100% bir hil ekanligin ko'rdik. Olingan natijani quyidagi diagrammada ko'rsatilgan.



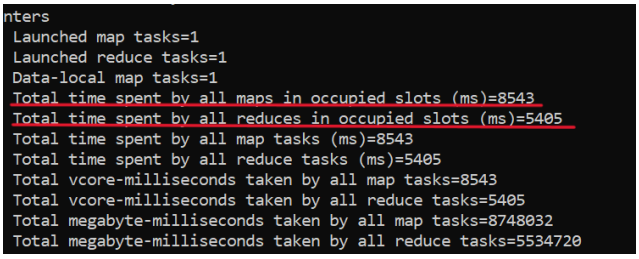
1-diagramma. Katta hajmli ma’lumotdagi so’zlarning chastotaviy tahlili

So’zlarning chastotaviy tahlilidan tashqari katta hajmli ma’lumotni qayta ishlash uchun sarflangan vaqt ham biz katt ahamiyatga ega. Chunki asosiy maqsadimiz qayta ishlash jarayoni tezlashtirishdan iborat. Quyidagi rasmda esa Java Corega asoslangan dasturda, biz tanlagan katta hajmli ma’lumotni qayta ishlash uchun sarflangan vaqtni ko’rishimiz mumkin.



4-rasm. Java Corega asoslangan qayta ishlash uchun sarflangan vaqt

Quyidagi rasmda esa Hadoop yordamida parallel hisoblashlarga asoslangan qayta ishlash uchun sarflangan vaqt va boshqa resurslarni ko’rishimiz mumkin.



5-rasm. Hadoop orqali parallel qayta ishlash uchun sarflangan vaqt.

XULOSA

Katta hajmli ma’lumotni parallel hisoblashlar yordamida qayta ishlash mavzusiga bag’ishlangan ushbu maqolani yozish jarayonida biz quyidagilarni amalga oshirdik:

- Katta hajmli ma’lumotlar va ularni qayta ishlashga doir adabiyotlarni tahlil qildik

- Apache Hadoop dasturini kompyuterimizga o’rnatdik va sozlab oldik
- Katta hajmli ma’lumotni topib, uni o’zimizga zarur formatga o’tkazdik
- Katta hajmli ma’lumotni taqsimlangan fayl tizimlarida saqlab oldik
- Katta hajmli ma’lumotni Java Core asoslangan ananaviy usulda qayta ishlab, natija oldik
- Katta hajmli ma’lumotni parallel hisoblash yordamida qayta ishlab, natijalar oldik
- Olingan natijalarni qiyosiy solishtirish asosida xulosalar chiqardik.

Ushbu o’tkazilgan tajribaning natijalariga asoslanib quyidagilarni xulosa qilish mumkin:

- Katta hajmli ma’lumotni Java Core asoslangan ananaviy usulda qayta ishlash mumkin, lekin juda ko’p hisoblashlarni va juda ko’p vaqtni talab qiladi;
- Katta hajmli ma’lumotni Hadoop yordamida parallel qayt ishlash mumkin, bu juda samarali va bu hisoblashlar kam vaqt talab qiladi;
- Ayni bir hil topshiriqni Java Core asoslangan ananaviy usuldagi qayta ishlash va Hadoop yordamida parallel qayt ishlashdan bir hil natida olish mumkin, lekin hisoblash uchun sarflangan vaqt bo’yicha katta farq mavjud;
- Hadoop yordamida parallel qayt ishlash uchun sarflangan umumiy vaqt 13 948 ms;
- Java Core asoslangan ananaviy usuldagi qayta ishlash uchun sarflangan vaqt 177 769 ms;
- Hisoblash vaqti bo’yicha Hadoopga asoslangan parallel hisoblash ananviy hisolashdan taxminan 13 marta tezroq ishlaganini ko’rishimiz mumkin.

REFERENCES

1. Onay, Ceylan; Öztürk, Elif "A review of credit scoring research in the age of Big Data". Journal of Financial Regulation and Compliance. . 2018 – C.382–405.
2. Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas Prem Prakash Jayaraman, Teh Ying Wah, Samee U. Khan. Big Data Reduction Methods: A Survey. Data Sci. Eng. (2016)
3. "Measuring the Business Value of Big Data | IBM Big Data & Analytics Hub". Www.ibmbigdatahub.com. 2021.
4. . Kitchin, Rob; McArdle, Gavin. "What makes Big Data, Big Data? Exploring the characteristics of 26 datasets".2016 Big Data & Society. 3 (1):
5. Алексеева И.Ю. Искусственный интеллект и рефлексия над знаниями. // ―Философия науки и техники‖: журнал 1991 №9, с. 44-53.

6. Urazmatov, T.Q.,Nurmetova, B.B.,Kuzibayev, X.S. Analysis of big data processing technologies. IOP Conference Series: Materials Science and Engineering, 2020, 862(4), 042006.
7. Urazmatov, T.Q.,Sh Kuzibayev, X. MapReduce and Apache spark: Technology analysis, advantages and disadvantages Journal of Physics: Conference Seriesthis link is disabled, 2022, 2373(5), 052008.
8. Ilhombekovich, S.B.,Kuzibayev K.S.,Xakimovna, A.G. Calculation of Synaptic Weights in Neuroexpert Systems International Conference on Information Science and Communications Technologies: Applications, Trends and Opportunities, ICISCT 2021, 2021.