PARALLEL HISOBLASH VA APACHE SPARK ASOSIDA KATTA HAJMDAGI MATNLARNI PUNKTUATSION TAHLIL QILISH

Maqsud Siddiqovich SHARIPOV
texnika fanlari nomzodi
Urganch davlat universiteti
Urganch, O‘zbekiston
maqsbek72@gmail.com
Xushnudbek Saylboyevich ADINAYEV
ўқитувчи
Urganch davlat universiteti
Urganch, O‘zbekiston
hushnudbek.adinaev@gmail.com

Annotatsiya

Zamonaviy tabiiy tilni qayta ishlash (NLP) vazifalaridan biri – matnni punktuatsion tahlil qilish bo‘lib, katta hajmli matnlar bilan ishlashda samarali hisoblash usullariga ehtiyoj yuqori. Ushbu tadqiqotda punktuatsion tahlil dasturini turli hisoblash arxitekturalarida – oddiy ketma-ket (Sequential, 1 CPU), parallel (CPU + GPU) va taqsimlangan (Spark klasteri) holda bajarib, ularning unumdorligi taqqoslangan. Tajribalar shuni ko‘rsatadiki, Spark klaster muhitida ma’lumotlarni qayta ishlash tezligi oddiy ketma-ket (sequential) hisoblashga nisbatan bir necha baravar yuqori bo‘lib, ayniqsa ma’lumot hajmi oshganda samaradorlik sezilarli darajada ortadi.

Tayanch so‘zlar: parallel hisoblash, Apache Spark, punktuatsiya, NLP, Big Data.

ПУНКТУАЦИОННЫЙ АНАЛИЗ БОЛЬШИХ ТЕКСТОВ НА ОСНОВЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ И APACHE SPARK

Максуд Сиддикович ШАРИПОВ
Кандидат технических наук
Ургенчский государственный университет
Ургенч, Узбекистан
maqsbek72@gmail.com
Хушнудбек Сайлбоевич АДИНАЕВ
преподователь
Ургенчский государственный университет
Ургенч, Узбекистан
hushnudbek.adinaev@gmail.com

Аннотация

Одной из современных задач обработки естественного языка (NLP) является пунктуационный анализ текста, при работе с большими объёмами данных возникает высокая потребность в эффективных методах вычислений. В данном исследовании программа пунктуационного анализа была выполнена на различных вычислительных архитектурах — последовательной (Sequential, 1 CPU), параллельной (CPU + GPU) и распределённой (кластер Spark) — с целью сравнения их производительности. Эксперименты

показали, что скорость обработки данных в кластерной среде Spark во много раз выше, чем при последовательных вычислениях, особенно при увеличении объёма данных эффективность значительно возрастает.
Ключевые слова: параллельные вычисления, Apache Spark, пунктуация, NLP, большие данные.

Tinish belgilarini tahlil qilish bilan dunyoning ko‘pchilik olimlari shug‘ullangan, jumladan, yaqin vaqtgacha tinish belgilari nazariy va hisoblash tilshunosligining aksariyat tadqiqotchilari tomonidan e’tibordan chetda edi. Bu mavhum muammo uchun ixcham, rasmiy asosning yo‘qligi bilan bog‘liq. Biroq, tinish belgilari yozma tilning orfografik tarkibiy qismi ekanligini esga olsak, tinish belgilariga oid tadqiqotlar oqilona ma’noga ega ekanligini ko‘ramiz. Shunga ko‘ra, so‘nggi o‘n yillikda mavzuga qiziqish ortdi, chunki tinish belgilarini hisobga olmasdan yozma tilni to‘liqroq tushunish va qayta ishlash mutlaqo mumkin emasligi tushunildi. Tinish belgilari dastlab yozma matnda intonatsiyani aks ettirish vositasi sifatida ixtiro qilingan bo‘lsa-da, endi u lingvistik “Tinish belgilari alohida tizim” dir [1].

Daniya tilidagi noto‘g‘ri vergullarni aniqlashni o‘rganish uchun Brill teggeridan (Brill 94,95) foydalanish bo‘yicha tadqiqotlarni tasvirlaydi. 600 000 so‘zdan iborat bo‘lgan nutqning bir qismi yorliqli korpus bo‘yicha o‘qitilgan tizim 91% aniqlik va 77% eslab qolish bilan noto‘g‘ri vergullarni aniqlaydi. Tizim matnga tasodifiy vergul qo‘yish orqali ishlab chiqilgan, ular noto‘g‘ri deb belgilangan, asl vergul esa to‘g‘ri deb belgilangan [2].

So‘nggi yillarda Big Data tushunchasi zamonaviy axborot texnologiyalarida markaziy o‘rinni egalladi [3]. Internet tarmoqlari, ijtimoiy tarmoqlar va IoT qurilmalari tomonidan yaratilayotgan katta hajmdagi matn va multimedia ma’lumotlari ularni qayta ishlashda samarali vositalarni talab qiladi. An’anaviy ketma-ket hisoblash usullari bu talablarni qondira olmaydi. Shu sababli parallel va taqsimlangan hisoblash tizimlari, jumladan Apache Spark, ilmiy va amaliy loyihalarda keng qo‘llanilmoqda [4].

Apache Spark – bu Big Datani qayta ishlash uchun mo‘ljallangan ochiq manbali framework. U dastlab 2009-yilda Kaliforniya universiteti (Berkeley) AMPLab laboratoriyasida ishlab chiqilgan va 2010-yilda Apache Software Foundationning ochiq manbali loyihasiga aylangan (Bichutskiy, 2015; Carlos Hinojosa, 2016). Sparkning asosiy xususiyati – bu xotirada taqsimlangan massiv ma’lumotlarni tahlil qilish bo‘lib, u dastur bajarilish tezligini sezilarli darajada oshiradi.

II. Apache Spark asosida parallel hisoblash. Apache Spark ochiq kodli hisoblash tizimi bo‘lib, katta hajmdagi ma’lumotlarni xotirada qayta ishlash imkoniyatiga ega. Spark arxitekturasi uchta asosiy qismdan iborat:

Driver program – asosiy boshqaruvchi modul, RDD va DataFrame obyektlarini yaratadi;

Cluster manager – resurslarni taqsimlaydi va ish jarayonlarini boshqaradi;

Worker nodes – ma’lumotlarni amalda qayta ishlovchi tugunlar.

Sparkda uchta asosiy ma’lumot strukturasi mavjud: RDD (Resilient Distributed Dataset), DataFrame va Dataset [5]. Bu strukturalar yordamida NLP jarayonlari – tokenizatsiya, stop-so‘zlarni olib tashlash, punktuatsiyani tiklash kabi vazifalar parallel tarzda bajariladi.

Sparkning MLlib kutubxonasi va Pipeline API yordamida mashinaviy o‘rganish jarayonlari bir nechta bosqichlarda ketma-ket, ammo parallel tarzda amalga oshiriladi.

III. Natijalar. A. Tajriba muhiti:

Oddiy tizim (Sequential): Intel i7-12700H, 16 GB RAM, 1 TB SSD.

Spark klasteri (Parallel): 5 ta kompyuter, har biri GeForce RTX GPU, 16 CPU, 32 GB RAM, 1 TB SSD.

B. Ma’lumotlar:

Matn korpuslari 1 MB, 5 MB, 10 MB, 15 MB, 20 MB, 25 MB, 30 MB, va 35 MB hajmlarda tayyorlandi.

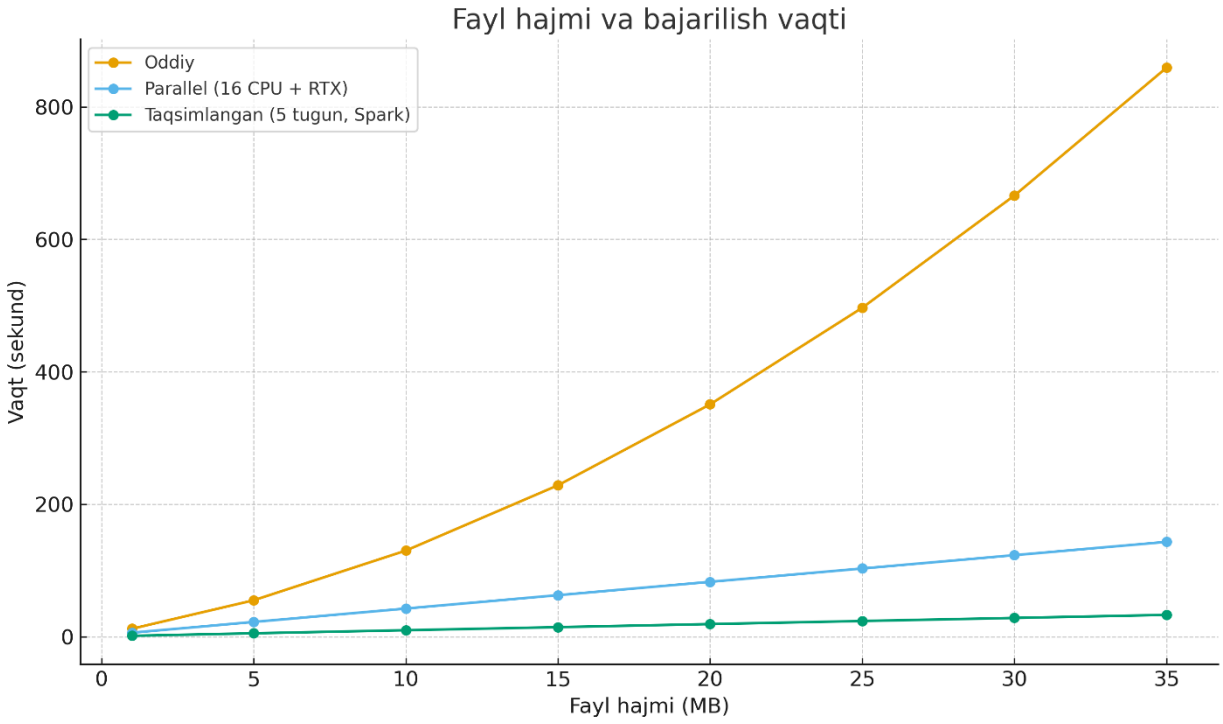
C. Natijalar jadvali:

1-jadval

Fayl hajmi (MB)	**Oddiy**	**Parallel (16 CPU + RTX)**	**Taqsimlangan (5 tugun, Spark)**
1	12	6	1.38
5	55	22.26	5.12
10	130	42.5	9.77
15	228.61	62.68	14.42
20	350.83	82.83	19.05
25	496.67	102.97	23.68
30	666.11	123.09	28.31
35	859.17	143.19	32.93

D. Tahlil

1-rasmdan ko‘rinib turibdiki, oddiy hisoblash ma’lumot hajmi ortishi bilan chiziqli ravishda sekinlashadi, Spark esa taqsimlangan hisoblash tufayli 3–4 baravar tezroq ishlaydi. Shu bois Spark NLP kabi ko‘p bosqichli algoritmlar uchun eng maqbul platforma hisoblanadi.



1-rasm. Natijalarning solishtirma tahlil grafigi.

Shunday qilib, katta hajmdagi o‘zbek tili matnlarni punktuatsion tahlil qilish vazifasi tahlil qilganimizda fayl hajmi oshgani sayin bajarilish vaqti ham oshadi. Oddiy (ketma-ket) bajarilishi juda sekin ishlaydi – fayl hajmi 35 MB bo‘lganda bajarilish vaqti 800 soniyadan oshadi. Parallel (16 CPU + RTX) usulda ishlash sezilarli tezlik beradi – ayni fayl hajmida 150 soniyadan biroz ortiq vaqt ketadi. Eng tezkor usul esa Spark asosida 5 tugunda taqsimlangan hisoblash bo‘lib, fayl hajmi oshsa ham bajarilish vaqti juda past (35 MB da ham 30 soniyadan kam) bo‘ladi. Bu shuni ko‘rsatadiki, katta fayllarga ishlov berishda parallel va taqsimlangan (Spark) texnologiyalardan foydalanish samaradorlikni bir necha baravar oshiradi. Shunday qilib, Spark katta hajmdagi matnlarni qayta ishlashda samarali vosita bo‘lib, NLP sohasida keng qo‘llanishi mumkin.

FOYDALANILGAN ADABIYOTLAR:

1. M. Bayraktar, B. Say, and V. Akman, “An analysis of English punctuation: the special case of comma,” *International Journal of Corpus Linguistics*, vol. 3, Jul. 1998, doi: 10.1075/ijcl.3.1.03bay.
2. D. Hardt, “Comma checking in Danish,” 2001.
3. Wigan, M., & Clarke, R. (2005). Big Data: Issues and Challenges. *Journal of Information Science*.
4. Zaharia, M., Chowdhury, M., Franklin, M., Shenker, S., & Stoica, I. (2010). Spark: Cluster Computing with Working Sets. *USENIX Conference on Hot Topics in Cloud Computing*.
5. Apache Spark Documentation. https://spark.apache.org/docs
6. M. Sharipov, J. Mattiev, J. Sobirov, and R. Baltayev, “Creating a Morphological and Syntactic Tagged Corpus for the Uzbek Language,” in *CEUR Workshop Proceedings*, 2022, pp. 93 – 98.
7. U. Salaev, E. Kuriyozov, and C. Gómez-Rodríguez, “SimRelUz: Similarity and Relatedness scores as a Semantic Evaluation Dataset for Uzbek Language,” in *1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages, SIGUL 2022 - held in conjunction with the International Conference*

on Language Resources and Evaluation, LREC 2022 - Proceedings, 2022, pp. 199 – 206.

8. M. Abdurashetona and I. O. Ismailovich, “Methods of Tagging Part of Speech of Uzbek Language,” in *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021*, 2021, pp. 82 – 85. doi: 10.1109/UBMK52708.2021.9558900.

9. M. Abdurashetona and U. Mokhiyakon, “Software Features and Linguistic Features of Uzbek Synonymizer,” in *Proceedings - 7th International Conference on Computer Science and Engineering, UBMK 2022*, 2022, pp. 171 – 175. doi: 10.1109/UBMK55850.2022.9919447.

10. Mengliyev, S. Shahabitdinova, S. Khamroeva, S. Gulyamova, and A. Botirova, “The morphological analysis and synthesis of word forms in the linguistic analyzer,” *Journal of Language and Linguistic Studies*, vol. 17, no. 1, pp. 558 – 564, 2021

11. K. Madatov, S. Bekchanov, and J. Vičič, “Dataset of Karakalpak language stop words,” *Data Brief*, vol. 48, 2023, doi: 10.1016/j.dib.2023.109111.

12. M. Sharipov and O. Sobirov, “Development of a Rule-Based Lemmatization Algorithm Through Finite State Machine for Uzbek Language,” in *CEUR Workshop Proceedings*, 2022, pp. 154 – 159.

13. M. Sharipov and O. Yuldashov, “UzbekStemmer: Development of a Rule-Based Stemming Algorithm for Uzbek Language,” in *CEUR Workshop Proceedings*, 2022, pp. 137 – 144.

14. Mengliev, E. Akhmedov, V. Barakhnin, Z. Hakimov, and O. Alloyorov, “Utilizing Lexicographic Resources for Sentiment Classification in Uzbek Language,” Jan. 2023, pp. 1720–1724. doi: 10.1109/APEIE59731.2023.10347765.

15. M. Sharipov, J. Mattiev, J. Sobirov, and R. Baltayev, ‘Creating a Morphological and Syntactic Tagged Corpus for the Uzbek Language’, CEUR Workshop Proceedings, vol. 3315, pp. 93–98, 2022.