MapReduce: Simplified Data Processing on Large Clusters

Maksim Norkin, ISIfm-13
maksim.norkin@ieee.org

I. Problema

Darbe analizuojama problema yra paskirstytų skaičiavimų įgyvendinimo įvairumas. Kiekvienas programuotojas gali įvairiai įgyvendinti savo pasirinktos problemos skaičiavimo mechanizmą. Prieš pradėdamas skaičiavimo operacijas, programuotojas turi pirmiausiai įvertinti kaip skaičiavimai bus paskirstyti tarp sistemų, kas nutinka, kuomet vienas skaičiavimas yra nutraukiamas, tačiau kiti skaičiavimai vykdomi toliau, kaip paskirstyti pačius duomenis tarp atskirų mazgų ir galiausiai kaip tai įvykdyti kuo greičiau.

Straipsnyje yra pateikiamas programavimo modelis ir pati metodika, kuri leidžia programuotojui iškarto aprašyti skaičiavimo algoritmą ir visiškai negalvoti apie prieš tai išvardintas problemas.

II. Darbo tikslas

Straipsnio tikslas yra pateikti sprendimą, kaip panaudojus bendrą programavimo modelį, galima labai lengvai skaičiavimo operacijas atlikinėti skirtinguose mazguose.

III. Uždaviniai

- Aprašyti programavimo modelį (2. paragrafas)
- Aprašyti MapReduce įgyvendinimą (3. paragrafas)
- Aptarti galimus patobulinimus (4. paragrafas)
- Pateikti įvykdymo greitaveikos matavimus (5. paragrafas)
- Parodyti įgyvendinimo pavyzdį Google sistemose (6. paragrafas)
- Aptarti panašius ir būsimus darbus (7. paragrafas)

IV. Vertinimas

- Turinys ir pavadinimas
  - Straipsnio pavadinimas tiesiogiai atitinka straipsnio turinį. Straipsnyje taip pat detaliai yra aprašomi taikymo pavyzdžiai.
- Aktualumas
  - Analizuojama problema yra labai aktuali. Kiekvieną dieną generuojamų duomenų kiekis didėja. Kompiuterio spartumas nespėja vyktis duomenų skaičiaus, todėl reikia ieškoti sprendimų kaip galima apdoroti didelius tera-baitinius duomenų kiekius. Šiuo metu tokios sistemos atviro kodo įgyvendinimą naudoja dauguma didelių kompanijų ir vos ne kiekvienas Sicilio slėnyje esantis start-up'as.
- Argumentavimas
  - Autoriai pateikia grafinius skaičiavimų įrodymus, pateikiamas konkretus veiksmų planas, kiekvienas žingsnis yra detaliai aprašomas ir apžvelgiamas. Grafiniai skaičiavimų įrodymai pateikia kiek laiko užtrunka duomenų persiuntimas iš vienos masino į kitą, lentelėje patiekiami skaitiniai duomenys tiek apie skaičiavimo greitaveiką, kiek vidutiniškai darbų buvo nutraukti dėl kažkokios kilmės gedimo ir kiek iš viso buvo įvykdytą užduočių per visą sistemos gyvavimo laikotarpį.
- Metodika
  - Modeliavimas, sisteminė analizė. Modeliuojamos yra užduotys, kurios yra perkeliamos iš standartinio įgyvendinimo iki MapReduce programavimo modelio įgyvendinimo. Sistemos analizė vykdoma pateikus detalius žingsnius kaip skaičiavimas yra vykdomas tarp skirtingų sistemos mazgų.
- Nuoseklumas
  - Pradedama nuo metodo aprašymo, tuomet keliamasi į bendrą algoritmo įgyvendinimą, aptariami galimi nesėkmių atvejai. Pateikiami realių problemų sprendimų pavyzdžiai. Supažindinama su jau esamu įgyvendinimu, pateikiami realūs skaičiavimų rezultatai.
- Problema, tikslas, uždaviniai, išvados

  - Iškelti uždaviniai išspręsti, detaliai išnagrinėjus kiekvieną iš jų.
- Bloomo taksonomija
  - Pasiekiamas Veiksmų plano lygmens. Straipsnis pradžioje labai trumpai pateikia informacija apie programavimo modelį, sujungia vienos programavimo kalbos vykdymo unikalumą, modelis perkeliamas didesniam masteliui.
- Stilius
  - Darbo stilius yra nuoseklus ir suprantamas kiekvieno, kas tik yra susidūręs su IT pobūdžio straipsniais
- Literatūra
  - Literatūros sąrašas stiprus. Cituojamai ACM, IEEE straipsniai, konferencijų prezentacijos.

V. Išvados

- Sėkmingas modelio panaudojimas Google sistemose (*MapReduce programming model has neem successfully used at Google for many different purposes*)
- Lengvas modelio panaudojimas (*The model is easy to use, even for programmers without experience with parallel and distributed systems*)
- Platus problemų sąrašas yra lengvai aprašomas per MapRecude skaičiavimus (*a large vriaty of problems are easily expressible as MapReduce compuations*)
- MapReduce algoritmo įgyvendinimas, kuris lengvai paskirstomas tarp tūkstančių skaičiavimo taškų (*An implementation of MapReduce that scales to large clusters of machines comprising thousands of machines*)
- Griežtas programavimo modelis leidžia lengviau skirstyti skaičiavimus tarp mašinų (*Restricting the programming model makes it easy to parallelize and distribute computations and to make such computations fault-tolerant*)
- Tinklo pralaidumas yra esminė problema (*the locality optimization allows us to read data from local disks, and writing a single compy of the intermediate data to local disk saves network bandwidth*)

Straipsnis “MapReduce: Simplified Data Processing on Large Clusters”, kurių autoriai Jeffrey Dean ir Sanjay Ghemawat, detaliai išanalizuotas. Pateikta analizuojama problema, darbo tikslas bei sąrašas uždavinių. Visi iškelti uždaviniai straipsnyje sėkmingai įvykdyti. Išvadose sekanti informacija yra pagrindžiama. Iš vertinimo požiūrio straipsnis yra vienas iš pavyzdinių.