

VILNIAUS UNIVERSITETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Srautinio apdorojimo sistemų balansavimas taikant mašininį mokymąsi

Balancing stream processing systems using machine learning

Mokslo tyrimasis darbas I

Atliko:	Vytautas Žilinas	(parašas)
Darbo vadovas:	Partn. Doc. Andrius Adamonis	(parašas)
Recenzentas:	Prof. Dr. Aistis Raudys	(parašas)

Vilnius – 2019

1. Tyrimas

1.1. Tyrimo aktualumas ir naujumas

Šiame magistriniame darbe bus nagrinėjami srautinio apdorojimo sistemų balansavimas ir bus bandoma tam pritaikyti mašininį mokymąsi. 2018 metų straipsnyje [VC18] pirmą kartą buvo panaudotas "Reinforcement learning" tipo modifikuotas "REINFORCE" algoritmas [SMS⁺99] siekiant automatiškai balansuoti "Apache Spark" srautinio apdorojimo sistemą. Šiame darbe bus bandomas kitas "Reinforcement learning" tipo algoritmas ir taip pat bus balansuojama "Heron" srautinio apdorojimo sistema.

Yra sukurta daug "Reinforcement learning" algoritmų, tad šiame darbe jie bus apžvelgti ir vienas iš jų bus pasirinktas ir pritaikytas išsikeltam uždaviniui. Algoritmas bus pasirinktas pagal tai, kuris bus tinkamiausias srautinių apdorojimų sistemų balansavimui.

Darbe nagrinėjama "Heron" srautinio apdorojimo sistema, kuri 2016 metais buvo išleista "Twitter", kaip alternatyva jau esamai "Apache Storm" srautinio apdorojimo sistemai [Ram16]. "Apache Spark" skiriasi nuo "Apache Storm" ir "Heron" tuo, kad ji apdoroja duomenis ne srautais, o mikro-paketais [KKW⁺15]. Šiame darbe bus nagrinėjamas "Heron", kadangi tai yra naujesnis srautinio apdorojimo variklis nei "Apache Storm" ir todėl, kad jo duomenų apdorojimo būdas yra kitoks nei "Apache Spark" kuris buvo naudojamas [VC18] straipsnyje.

2. Darbas

2.1. Darbo tikslas

Ištirti mašininio mokymosi tinkamumą srautinio apdorojimo sistemų balansavimui.

2.2. Uždaviniai

1. Pasirinkti srautinio apdorojimo sistemos metrikas, kurios bus naudojamos balansavimui.
2. Išanalizuoti esamus "Reinforcement Learning" algoritmus ir pasirinkti tinkamą tyrimui.
3. Pritaikyti pasirinktą mašininio mokymosi algoritmą srautinio apdorojimo sistemos balansavimui.
4. Straipsnyje [VC18] pateikto bandymo pritaikymas "Heron" srautinio apdorojimo sistemai.
5. Atlikti eksperimentą ir palyginti rezultatą su alternatyvomis - standartinė konfigūracija, reaktyvus balansavimas.

2.3. Laukiami rezultatai

1. Pritaikant skirtingas apkrovas srautinio apdorojimo sistemoms, surinktos ir surikiuotos metrikos, reikalingos "Heron" srautinio apdorojimo sistemos balansavimui.
2. Pasirinktas "Reinforcement Learning" mašininio mokymosi algoritmas.
3. Pasirinktas mašininio mokymosi algoritmas pritaikytas "Heron" srautinio apdorojimo sistemų balansavimui.
4. Atliktas eksperimentas naudojant sukurtą balansavimo implementaciją ir srautinio apdorojimo sistemos dirbtines apkrovas.
5. Išbandytas [VC18] straipsnyje taikomas algoritmas.
6. Eksperimento rezultatai palyginti su standartine konfigūracija ir reaktyviu balansavimu.

Literatūra

- [KKW⁺15] Holden Karau, Andy Konwinski, Patrick Wendell ir Matei Zaharia. *Learning spark: lightning-fast big data analysis.* ” O’Reilly Media, Inc.”, 2015.
- [Ram16] Karthik Ramasamy. Open sourcing twitter heron, 2016.
- [SMS⁺99] Richard S. Sutton, David McAllester, Satinder Singh ir Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS’99, p. 1057–1063, Denver, CO. MIT Press, 1999. URL: <http://dl.acm.org/citation.cfm?id=3009657.3009806>.
- [VC18] Luis M. Vaquero ir Felix Cuadrado. Auto-tuning distributed stream processing systems using reinforcement learning, 2018. arXiv: 1809.05495 [cs.DC].