



Студенттер мен жас ғалымдардың
«ҒЫЛЫМ ЖӘНЕ БІЛІМ - 2018»
XIII Халықаралық ғылыми конференциясы

СБОРНИК МАТЕРИАЛОВ

XIII Международная научная конференция
студентов и молодых ученых
«НАУКА И ОБРАЗОВАНИЕ - 2018»

The XIII International Scientific Conference
for Students and Young Scientists
«SCIENCE AND EDUCATION - 2018»



12th April 2018, Astana

**ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Л.Н. ГУМИЛЕВ АТЫНДАҒЫ ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ**

**Студенттер мен жас ғалымдардың
«Ғылым және білім - 2018»
атты XIII Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ**

**СБОРНИК МАТЕРИАЛОВ
XIII Международной научной конференции
студентов и молодых ученых
«Наука и образование - 2018»**

**PROCEEDINGS
of the XIII International Scientific Conference
for students and young scholars
«Science and education - 2018»**

2018 жыл 12 сәуір

Астана

УДК 378

ББК 74.58

Ғ 96

Ғ 96

«Ғылым және білім – 2018» атты студенттер мен жас ғалымдардың XIII Халықаралық ғылыми конференциясы = XIII Международная научная конференция студентов и молодых ученых «Наука и образование - 2018» = The XIII International Scientific Conference for students and young scholars «Science and education - 2018». – Астана: <http://www.enu.kz/ru/nauka/nauka-i-obrazovanie/>, 2018. – 7513 стр. (қазақша, орысша, ағылшынша).

ISBN 978-9965-31-997-6

Жинаққа студенттердің, магистранттардың, докторанттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.

В сборник вошли доклады студентов, магистрантов, докторантов и молодых ученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 378

ББК 74.58

ISBN 978-9965-31-997-6

©Л.Н. Гумилев атындағы Еуразия
ұлттық университеті, 2018

студенттер, магистранттар, докторанттар мен жас мамандарға ғылыммен айналысуға жоғары деңгейде жағдай жасауға болады деген тұжырымға келдік.

Қолданылған әдебиеттер тізімі

1. Мырзабеков А.В., Сьюторенко О.В. «Қазақ сандық кітапханалары» бағдарламасы // Электронды кітапханалар: ғылыми. электрон. - 1998. - Т. 1. - Шығарылым. 1. - Кіру режимі: <http://www.elbib.kz/content/journal>

2. Проект «Гутенберг» [Электронный ресурс]: [сайт проекта]. – [Б. м., 20-]. – Режим доступа: <http://www.gutenberg.org> .

PYTHON БАҒДАРЛАМАЛАУ ТІЛІНДЕ ПАРАЛЛЕЛЬДІ ЕСЕПТЕУЛЕР ЖҮРГІЗУ

Жоланова Қарлығаш Бақтиярқызы

Л.Н. Гумилев атындағы Еуразия ұлттық университетінің
«5В011100- Информатика» мамандығының студенті, Астана, Қазақстан
Ғылыми жетекшісі – Зулпыхар Ж.Е.

XXI ғасырдан келе жатқан параллель есептеулер күрделі және қол жетімсіз болатын. Python тілі мен қоса көптеген платформалар мен тілдер параллелизацияны жүзеге асыруға мүмкіндік береді. Python көптеген тапсырмаларды орындауға арналған көптеген құралдарды ұсынады: көп дәрежелі, көп процедуралы. Қысқаша айтқанда, математикалық есептеулер үшін көп функциялы мультиқауіпсіздікті қолдануға болады.

Python + NumPy + Matplotlib байламы, ғылыми есептер мен алгоритмдердің жылдам прототипін жасау үшін ең тиімдісі. Бір циклде қарапайым дәйекті іздеуді қолдансаңыз, бұл шешім ұзақ уақыт бойы шешіледі. Көп ядролы процессорға ие болу осындай міндеттерді оңай параллелдеуге және күшті күшпен шешуге болады.

Python-да параллель есептеулерді жүзеге асыру үшін multiprocessing модульді қолданылады. Барлық процессорлық ядроларды жүктеу арқылы 5000,000-нан 6 000 000-ға дейінгі сандардағы қарапайым сандарды санау. Кәдімгі CPython жүйесіндегі GIL әдеттен тыс пішін үлгісін мүлдем тиімсіз етеді, өйткені біз параллель есептеулер үшін процестердің пулын стандартты мультипроцессорлық модульді қолданамыз. Workers саны таңдау - Егер оның саны аз болса, мәселе баяу шешіледі. Үлкен сан болса жылдан шешіледі. Ол үшін алдымен worker-ге бөлнім. Worker саны 6 немесе оданда көп worker-ке бөлуге болады. Тапсырма менеджері барлық процессорлық ядролардың 100% жуығын көрсетеді.

Питонда жіптер POSIX Сызбалары арқылы жүзеге асыры. Сондай-ақ, GIL (Global Interpreter Lock) - бұл жіптерде. Тарихи, аудармашының жаһандық кеңістікте деректер құрылымдары ең қауіпсіз емес. Сондықтан, әзірлеушілер бір мезгілде жұмыс істейтін ағындардың санын шектеуге шешім қабылдады. Яғни, сол уақытта ол GIL маркерді түсіреді тек бір жіп өңдеуге болады. Әрбір 100-дікте аудармашы құлыпты босатып, басқа бір ағынды орындауға мүмкіндік береді.

Осылайша, қазіргі уақытта python-дағы многопрограммаларды енгізу кезінде біз бір ғана жіпті тек бір ғана CPU-да орындауға болады. Өкінішке орай, python 3-ні C-shnoy-де іске асыру кезінде бәрі өзгеріссіз қалады.

Атап айтқанда, біздің жағдайда, сандардың диапазонын тең бөліктерге бөліп, әрқайсысы ауқымның басы мен аяғынан сұрай алады, содан кейін барлық Worker-лер берілген диапазондарда табылған қарапайым нөмірлерді қайтарып, оларды қосып, күте тұрыңыз. Дегенмен, жұмысты тең бөлікке бөлуге болмайды. Атап айтқанда, қарапайым сандарды іздегенде күрделілік суб-ауқымның бастапқы мәнімен айтарлықтай артады.

Соңғы уақытта бірнеше қиындықтармен бетпе-бет келгенде, бір жіптегі тапсырма ең оңтайлы шешім емес. Мысалы, gui қосымшасында ауыр тапсырманы өңдеу кезінде, бағдарлама интерфейсі баяу жаңартылады, содан кейін орындалатын тапсырма үшін

қосымша ағынды іске қосу пайдалы және gtk бағдарламасы сурет салу үшін жауап беретін өзекті ағыны қалдыруы керек, мысалы, прогресс жолағы. Басқа міндет xml-файлдар тобын талдау және дерекқорға нәтиже енгізу болды - әрқайсысы жекелеген файлдармен жұмыс істейтін бірнеше процесті іске қосу ыңғайлы.

Питонда параллелділікпен жұмыс істеу үшін, екі кіріктірілген модуль бар: **1.multiprocessing** - бұл ағынды модульге ұқсас API пайдаланып, уылдырықтау үдерістерін қолдайтын пакет. Көптеген өңдеу пакеті жергілікті және қашықтағы параллельділікпен қамтамасыз етеді, жіптердің орнына субпроцесстерді пайдалану арқылы ғаламдық ауызша құлыптауды тиімді түрде қосады. Осыған байланысты, көп функциялы модуль бағдарламалаушыға берілген машинада бірнеше процессорды толығымен пайдалану мүмкіндігін береді. Ол Unix және Windows жүйелерінде жұмыс істейді.

Мультипроцессорлық модуль, сондай-ақ, ағынды модульде аналогтары жоқ API-ді енгізеді. Бұған ең қарапайым мысал болып табылады, ол көптеген кіріс мәндері бойынша функцияның орындалуын параллелдеудің ыңғайлы әдістерін ұсынады, бұл деректерді процестер (деректер параллелизмі) арқылы таратады. Төмендегі мысал модульдегі осындай функцияларды анықтаудың жалпы практикасын көрсетеді, сонда бала процестері осы модульді сәтті импорттай алады.

```
from multiprocessing import Pool
```

```
def f(x):  
    return x*x
```

```
if __name__ == '__main__':  
    with Pool(5) as p:  
        print(p.map(f, [1, 2, 3]))  
    нәтиже [1, 4, 9]
```

Көп функциялы процесте Процестің объектісін құру және оның start () әдісін шақыру арқылы процестер пайда болады. Процесс үрдістің API интерфейсіне сәйкес келеді.

Multiprocessing платформа түріне қарай, процесті бастаудың үш тәсілі бар. Бұл бастау әдістері

- **Spawn (ұрықтандыру)**

Ұрықтандыру процесі ол жаңа python аудармашы процесін бастайды. Бұл процесс объектілері run () әдісін іске қосу үшін қажетті ресурстарды ғана иеленеді. Атап айтқанда, ата-аналық процестің қажетсіз файлдық дескрипторлары мен тұтқалары қол жетімді болмайды. Бұл әдісті қолдана отырып процесті іске қосу форманың немесе форсунктың көмегімен жүзеге асады.

Unix және Windows жүйесінде қол жетімді.

- **Fork (шанышқы)**

Бұл процесс os.fork () тілін Python аудармашы үшін пайдаланады. Келесе процес басталса, алдыңғы процесіне тиімді болады. Ата-ананың барлық ресурстары бала процесі арқылы мұраға алады.

- **Forkserver (шабуылшы)**

Бағдарлама басталған кезде және серверлерді бастау әдісін таңдайды, сервер процесі басталады. Содан кейін, жаңа процесс қажет болғанда, ата-ана процесі серверге қосылып, жаңа процесті жүргізуді сұрайды. Форк сервер процесі бірыдролы, сондықтан os.fork () функциясын пайдалану қауіпсіз болып табылады. Қажет емес ресурстар мұра болып табылмайды

2.threading- Бұл модуль төменгі деңгейдегі _thread модулінің жоғары деңгейлі ағындық интерфейстерді құрастырады.

```
mydata=threading.local()
mydata.x=1
```

Бұл модуль келесі функцияларды анықтайды:

- ***threading.active_count*** ()

Қазіргі уақытта тірі нысандардың санын қайтарыңыз. Қайтарылған санау тізіммен қайтарылған тізім ұзындығына тең.

- ***threading.current_thread*** ()

Қоңырау шалушының басқару элементіне сәйкес келетін ағымдағы Thread нысанын қайтарыңыз. Егер қоңырау шалушы басқару элементі ағымдық модуль арқылы жасалмаса, шектеулі функционалдығы бар күлгін жіп нысаны қайтарылады.

- ***threading.get_ident*** ()

Ағымдағы 'ағымдық идентификаторды' қайтарыңыз. Бұл нөлден тыс бүтін сан. Оның мәні тікелей мағынасы жоқ; ол пайдаланылатын сиқырлы куки ретінде қарастырылады, мысалы: ағынға тән деректердің сөздік индексіне индекстеу. Туыстық идентификаторлар ағын пайда болғанда және басқа ағын пайда болғанда қайта өңделуі мүмкін.

Қолданылған әдебиеттер тізімі:

1. Электрондық ресурс <https://docs.python.org/3/library/threading.html>
2. Электрондық ресурс <https://docs.python.org/dev/library/multiprocessing.html>
3. “Параллельные вычисления и многопоточное программирование “, Биллиг В.А, М.: НОУ "Интуит",2016-310с.
4. " Parallel Programming with Python ", janpalach, 2014с.

ӘОЖ 004.4

ANDROID ЖҮЙЕСІНДЕ «INFO teacher»ҚОСЫМШАСЫН ЖАСАУ

Зиятов Акбар Мырахметович

Қостанай мемлекеттік педагогикалық институтының «Информатика»

мамандығының 4-курс студенті, Қостанай, Қазақстан

Ғылыми жетекшісі - А. Айтбенова

Бұрынғы жылдардың телефондарының негізгі міндеттері қоңыраулар шалу, қабылдау мен SMS жазу болса, қазіргі кезде бұл міндеттерге интернет – ресурстармен жұмыс, әуен тыңдау, бейне түсіру, ойындар мен қосымшалар қолдану сияқты көптеген мүмкіндіктермен толықтырылды. Енді мобильді телефондардың орнына жаңартылған, жаңа функцияларға ие смартфондар кеңінен қолданылуда. Дербес мобильді құрылғылар (смартфондар, планшеттер, дербес қалта компьютерлері) қазіргі замандағы әлемнің ажырамас бөлігі болып табылады. Мобильді құрылғылардың әрбір пайдаланушысы тілдесу үшін аса күшті құралға қол жеткізді. Смартфондардың нарығы жылдам дамып келеді, тіпті еңгезердей серіктестіктердің барлығы оның өзгерістеріне үлгеретін емес.

Басты мобильдік құралдардың қарқынды дамуы 2001 жылы Nokia атты компанияның алғаш қадамынан басталған, сол кезден бастап көптеген компаниялар өздерінің тұнғыш мобильдік бәйгесіне түскен болатын. Мобильдік қосымша негізінен өз тарихын бастамасын мобильдік ОЖ-нен шыққан кезден бастап тарихын жалғастырған.

Бүгінде мобильді құрылғылар арасында, негізгі тұтынушылар арасында танымал мына ОЖ жүйелер : IOS, Android, Windows phone.

Қазіргі кезде смартфондардың адам өміріне көмегі біз ойлағаннан әлде қайда көп тиіп жатыр. Android операциялық жүйесі мобильді құрылғылар үшін салыстырмалы түрде жаңа платформа болып табылады. Ол өзінің ашықтығының арқасында, жетілдіруге арналған тегін