

Дәріс 12. Конкуренттік программалау негіздері. Конкуренттік программалау артықшылықтары мен қиындықтары.

Дәрістің мақсаты: Студенттерде конкурентті программалаудың ерекшеліктері туралы түсінік қалыптастыру.

Дәрісті меңгеру нәтижесінде студенттер келесі қабілеттерге ие болады:

- Конкуренттік программалау ұғымын түсіну;
- Конкуренттік программалау артықшылықтарын түсіну;
- Конкуренттік программалау қиындықтарын түсіну.

Конкуренттілік бірнеше бағдарламаларды немесе бағдарламаның бірнеше бөліктерін бір мезгілде орындау мүмкіндігін сипаттайды. Компьютерлік бағдарламалауда қосымшада параллелизмді пайдалану қосымшаны әртүрлі ағындарда бір мезгілде (бір мезгілде) орындалатын бірнеше тәуелсіз процестерге бөле отырып, нақты көп міндеттілікті қамтамасыз етеді. Бұл ОП-ның бір ядросында немесе ОП-ның бірнеше ядролары қолжетімді болса, қатар болуы мүмкін. Бағдарламаның өткізу қабілеті (ОП есептеуді өңдейтін жылдамдық) және жылдамдығы тапсырманы асинхронды немесе қатар орындау есебінен жақсартылуы мүмкін. Бейнеконтентті ағынды жіберуді жүзеге асыратын қосымша параллель болып табылады, мысалы, бір мезгілде желіден сандық деректерді оқиды, оларды ашады және экрандағы көріністі жаңартады.

Конкуренттілік бұл ағындар қатар орындалады және бағдарламаның әр түрлі бөліктері бір мезгілде орындалуы мүмкін деген әсер қалдырады. Бірақ бір ядролы ортада бір ағынды орындау уақытша тоқтатыла тұрады және екіншісіне ауыстырылады. Компьютерлік бағдарламалауда бұл процесс параллелизм деп аталады.

Конкуренттілік - бұл өмірдің табиғи бөлігі, өйткені біз адамдар көп тапсырманы орындауға дағдыланғанбыз.

Электрондық поштаны кофе ішіп отырып оқи аламыз немесе сүйікті әнімізді тыңдай отырып тере аламыз. Қосымшада параллельді қолданудың негізгі себебі – ұлғайту өнімділік пен жауаптылық, және төмен кідіріске жету. Егер бір адам екі тапсырманы бірінен соң бірін орындайтын болса, екі адам бір уақытта сол тапсырманы орындағаннан гөрі көп уақыт қажет болады.

Қолданбалармен бірдей. Мәселе мынада, көптеген қосымшалар қол жетімді процессорлар арасында қажет тапсырмаларды біркелкі бөлу үшін жазылмайды. Компьютерлер көптеген түрлі салаларда қолданылады, мысалы, аналитика, қаржы, ғылым және денсаулық сақтау. Талданған мәліметтер саны жылдан-жылға артып келеді. Екі жақсы иллюстрация - Google және Pixar.

2012 жылы Google минутына 2 миллионнан астам іздеу сұрауын алды; 2014 жылы бұл сан екі еседен астам өсті. 1995 жылы Pixar компьютерде құрастырылған алғашқы «Toy Story» фильмін шығарды. Компьютерлік анимацияда көлеңкелену және жарықтандыру сияқты әр кескінге көптеген мәліметтер мен мәліметтер берілуі керек. Бұл ақпараттың барлығы секундына 24 кадр жылдамдығымен өзгереді. 3D фильмінде ақпараттың өзгеруін экспоненциалды арттыру қажет.

Toy Story-ді жасаушылар өз фильмдерін жасау үшін 100 қос процессорлы машиналарды пайдаланды, ал параллельді есептеуді қолдану өте қажет болды. Pixar құралдары Toy Story 2 үшін дамыды; компания сандық фильмдерді өңдеу үшін 1400 компьютерлік процессорларды қолданды, осылайша цифрлық сапа мен монтаждау уақытын едәуір жақсартты. 2000 жылдың басында Pixar-дың компьютерлік қуаты одан да артып, 3500 процессорға жетті. Он алты жылдан кейін толық анимациялық фильмді өңдеуге арналған компьютерлік қуат абсурдты 24000 ядроларға жетті. Параллельді есептеу қажеттілігі экспоненталық өсе береді.

N (кез келген сан сияқты) жұмыс істейтін ядролары бар процессорды қарастырайық. Бір ағынды қосымшада тек бір ядро жұмыс істейді. Бірнеше ағындарды орындайтын бірдей бағдарлама жылдамырақ болады және өнімділікке сұраныс өскен сайын N параллельді бағдарламалар болашаққа стандартты бағдарламалау моделін жасай отырып, өседі.

Егер сіз қосымшаны үйлестіре отырып ойластырылмаған көп ядролы машинада іске қосатын болсаңыз, компьютердің өнімділігін ысырап етесіз, өйткені ол процестердің тізбектелуіне байланысты қолданба компьютердің тек қуатының бір бөлігін пайдаланады. Бұл жағдайда, егер сіз Тапсырмалар менеджерін немесе кез-келген процессордың жұмысының есептегішін ашсаңыз, онда сіз тек бір ядроның жоғары деңгейде жұмыс істейтінін байқайсыз, мүмкін 100%, ал қалған барлық ядролар жеткіліксіз немесе бос тұр. Сегіз ядролы машинада бір мезгілде емес бағдарламаларды іске қосу ресурстарды жалпы пайдалану 15% төмен болуы мүмкін дегенді білдіреді.

Параллельді бағдарламалаудың бүгіні мен болашағы

Keңейтілген бағдарламаларды ұсыну үшін параллельдікті игеру қажетті дағдыға айналды. Компаниялар параллель код жазуды терең білетін инженерлерді жалдауға және инвестициялауға мүдделі. Шындығында, параллельді дұрыс есептеуді жазу уақыт пен ақшаны үнемдеуге мүмкіндік береді. Аз серверлермен қол жетімді есептеу ресурстарын пайдаланатын масштабталатын бағдарламаларды құру арзан өнімді сатып алу және қосу деңгейіне жету үшін жеткіліксіз пайдаланылатын жабдықты қосудан гөрі аз серверлерде бар. Сонымен қатар, көптеген жабдықтар көп техникалық қызмет көрсетуді және электр қуатын қажет етеді.

Бұл көп ағынды код жазуды үйренудің қызықты уақыты және сіздің бағдарламаңыздың функционалды бағдарламалау (FP) тәсілімен өнімділігін жақсарту пайдалы. Функционалды бағдарламалау - бұл есептеуді өрнектерді бағалау ретінде қарастыратын және өзгеретін күй мен өзгеретін мәліметтерден аулақ болатын бағдарламалау стилі. Өзгермейтіндік әдепкі болып табылады және фантастикалық композиция мен декларативті бағдарламалау стилі қосылғандықтан, FP қатарлас бағдарламалар жазуды қиындықсыз етеді.

Жаңа парадигмада ойлау біршама ыңғайсыз болса да, параллель бағдарламалауды үйренудің алғашқы қиындықтары тез азаяды, ал табандылық үшін сыйақы шексіз. Windows Task Manager бағдарламасын ашып, сіздің кодыңыз өзгергеннен кейін процессордың қолданылуы 100% -ке дейін өсетінін мақтан тұтып, сиқырлы және керемет нәрсе таба аласыз. Функционалды парадигманы қолдана отырып, жоғары масштабталатын жүйелерді жазуды үйреніп, сізге ыңғайлы болғаннан кейін, ретті кодтың баяу стиліне оралу қиын болады.

Параллельдік - бұл компьютерлік индустрияда үстемдік ететін келесі жаңашылдық және ол әзірлеушілердің бағдарламалық жасақтаманы қалай жазатындығын өзгертеді. Өнеркәсіптегі бағдарламалық жасақтамаға қажеттіліктердің эволюциясы және

оқшауламайтын интерфейстер арқылы пайдаланушының үлкен тәжірибесін беретін жоғары өнімді бағдарламалық жасақтамаға деген сұраныс параллельділік қажеттілігін арттыра береді. Жабдықтың бағытымен құлыптау кезінде параллелизм бағдарламалаудың болашағы екені анық.

Конкурентті программалаудың қиындықтары

Бір уақытта және параллель бағдарламалау берілген есептеулердің жылдамдығы мен жылдам орындалуы үшін пайдалы екені сөзсіз. Бірақ өнімділік пен реактивті тәжірибенің бұл бағасы өзіндік бағамен келеді. Тізбектелген бағдарламаларды қолдана отырып, кодтың орындалуы болжау мен детерминизмнің бақытты жолына түседі. Керісінше, көп тізбекті бағдарламалау дұрыстылыққа жету үшін адалдық пен күш-жігерді қажет етеді.

Сонымен қатар, бірнеше орындалу туралы бір уақытта айту қиын, өйткені біз дәйекті ойлауға дағдыланғанбыз.

Қатар бағдарламалар жазу оңай емес және бағдарламаны әзірлеу кезінде көптеген күрделі элементтерді ескеру қажет. Жаңа ағындар жасау немесе ағындар пулында бірнеше тапсырмаларды кезекке қою салыстырмалы түрде оңай, бірақ бағдарламаның дұрыстығын қалай қамтамасыз ету керек? Көптеген ағымдар жалпы деректерге үнемі хабарласқанда, оның тұтастығына кепілдік беру үшін деректердің құрылымын қалай қорғау керектігін қарастыру қажет. Ағын басқа ағындардың араласуынсыз жадының орнын атомарлы түрде жазуы және өзгертуі тиіс. Шындық дегеніміз, бағдарламалаудың императивтік тілдерінде немесе мәндері өзгеруі мүмкін (өзгертін айнымалылар) ауыспалы тілдерде жазылған бағдарламалар жадты синхрондау деңгейіне немесе қатар қолданылатын кітапханаларға карамастан, деректерді жарысуға әрдайым осал болады.