JavaScript негізінде параллельді есептеу

Ернар Тоқтар
Астана халқаралық университет

ӘОЖ: 004.42

АННОТАЦИЯ

Жобаның мақсаты: JavaScript дегі Web Workers технологиясы арқылы параллельді есептеулерді жүзеге асыру.

Жоба өзектілігі: JavaScript-дің бір ағынды орындау моделі ауқымды деректерді өңдеудің және күрделі есептеулердің тиімділігін шектейді.
Параллельді есептеулер концепциясы тапсырмаларды бір уақытта орындауға болатындай етіп тәуелсіз ішкі тапсырмаларға бөлуді қамтиды, осылайша жалпы өнімділікті арттырады. JavaScript дегі Web Workers — JavaScript кодын фондық ағында орындайтын, параллельді есептеу мүмкіндігін қамтамасыз ететін технология. Бірнеше жұмыс ағындарын жасау арқылы әзірлеушілер негізгі ағынға әсер етпестен параллельді есептеу тапсырмаларын орындай алады.

АННОТАЦИЯ

Цель проекта: Реализация параллельных вычислений с использованием технологии Web Workers на JavaScript.

Актуальность проекта: Модель однопоточного выполнения JavaScript ограничивает эффективность крупномасштабной обработки данных и сложных вычислений.
Концепция параллельных вычислений предполагает разделение задач на независимые подзадачи, чтобы их можно было выполнять одновременно, тем самым увеличивая общую производительность. Web Workers в JavaScript — это технологии параллельных вычислений, которые выполняют код JavaScript в фоновом потоке. Создавая несколько рабочих процессов, разработчики могут выполнять параллельные вычислительные задачи, не затрагивая основной поток.

ANNOTATION

Project objective: Implementation of parallel calculations using Web Workers technology in JavaScript.

The relevance of the project: JavaScript's single-threaded execution model limits the efficiency of large-scale data processing and complex computations.
The concept of parallel computing involves dividing tasks into independent subtasks so that they can be executed simultaneously, thereby increasing overall performance. Web Workers in JavaScript are parallel computing technologies that execute JavaScript code in a background thread. By creating multiple workflows, developers can perform parallel computing tasks without affecting the main thread.

1 Кіріспе:

1.1 Есептеу жылдамдығының қажеттілігі

Деректер масштабының жылдам өсуі және мәселелердің күрделене түсуі жағдайында бір компьютерді кең ауқымды жылдам есептеулердің қажеттіліктерін қанағаттандыру қиынға соғады.

Дәстүрлі компьютерлер әдетте бір ғана процессормен (яғни, орталық процессормен) жабдықталған. Есептеу жұмыстарына үлкен сұраныстарға тап болған кезде, бір компьютердің есептеу қуаты жеткіліксіз екені анық. Сондықтан, бұл мәселені шешу үшін параллельді есептеулер, қуатты есептеу парадигмасы пайда болды.

1.2 Параллельді есептеу

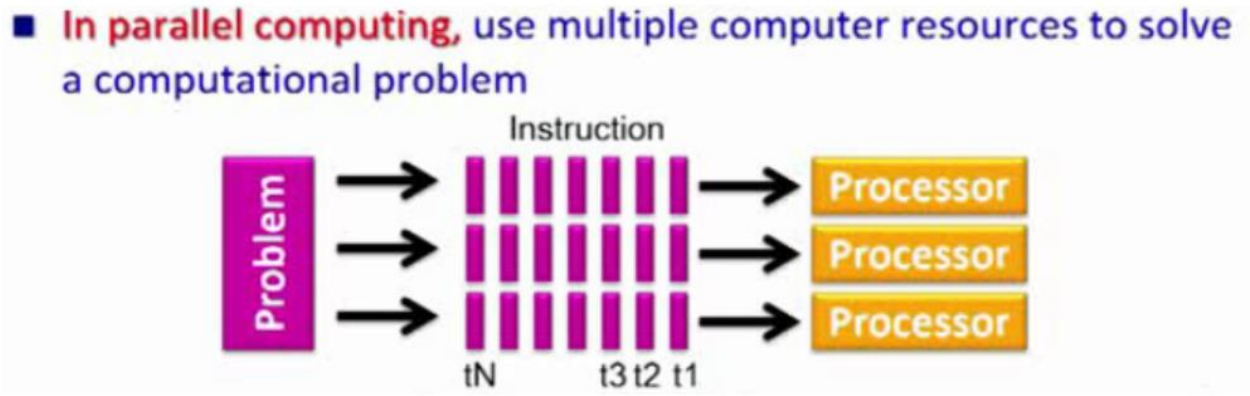
Қоғамдық әдебиеттерден табуға болатын ертерек анықтаманы Gottlieb және т.б. адамдар 1989 жылы «Жоғары өнімділік есептеулері» кітабында берген. Олар параллельді есептеулерді көптеген есептеулердің немесе есептеу процестерінің орындалуы бір уақытта орындалатын есептеулердің бір түрі деп санайды [1].

Параллельді есептеудің мақсаты – есептеулерді жылдамдату [2]. Негізгі ұғым: Параллельді есептеу деп есептеу мәселесін параллельді машинада бірнеше қосалқы тапсырмаларға ыдырату және оларды әртүрлі процессорларға тағайындау жатады. Әрбір процессор қосалқы тапсырмаларды параллель өңдеу үшін бір-бірімен ынтымақтасады, сол арқылы декомпозицияны жылдамдатады немесе үлкен есептерді шешеді.

Негізгі идея - бір есептеу мәселесін бірлесіп шешу үшін бірнеше процессорларды пайдалану, яғни әрбір процессор бүкіл есептеу тапсырмасының бір бөлігін дербес орындайды.

Қарапайым тілмен айтқанда, параллельді есептеулер - бұл бірнеше процессорлар бір уақытта үлкен күрделі мәселеден бөлінген бірнеше, кішірек есептеу тапсырмаларын орындайтын есептеу

архитектурасы. Параллельді есептеулер соңғы жылдары негізінен көп ядролы процессорлар түрінде компьютер архитектурасында басым парадигмаға айналды [3].



1-сурет. Параллельді есептеу диаграммасы

Параллельді есептеулердің анықтамасы бойынша параллельді есептеулер үш негізгі шартқа сай болуы керек [4]:

(1) Параллель машина. Параллельді машинада кем дегенде екі немесе одан да көп процессорлар бар, олар өзара байланыс желісі арқылы бір-бірімен қосылып, байланысады.

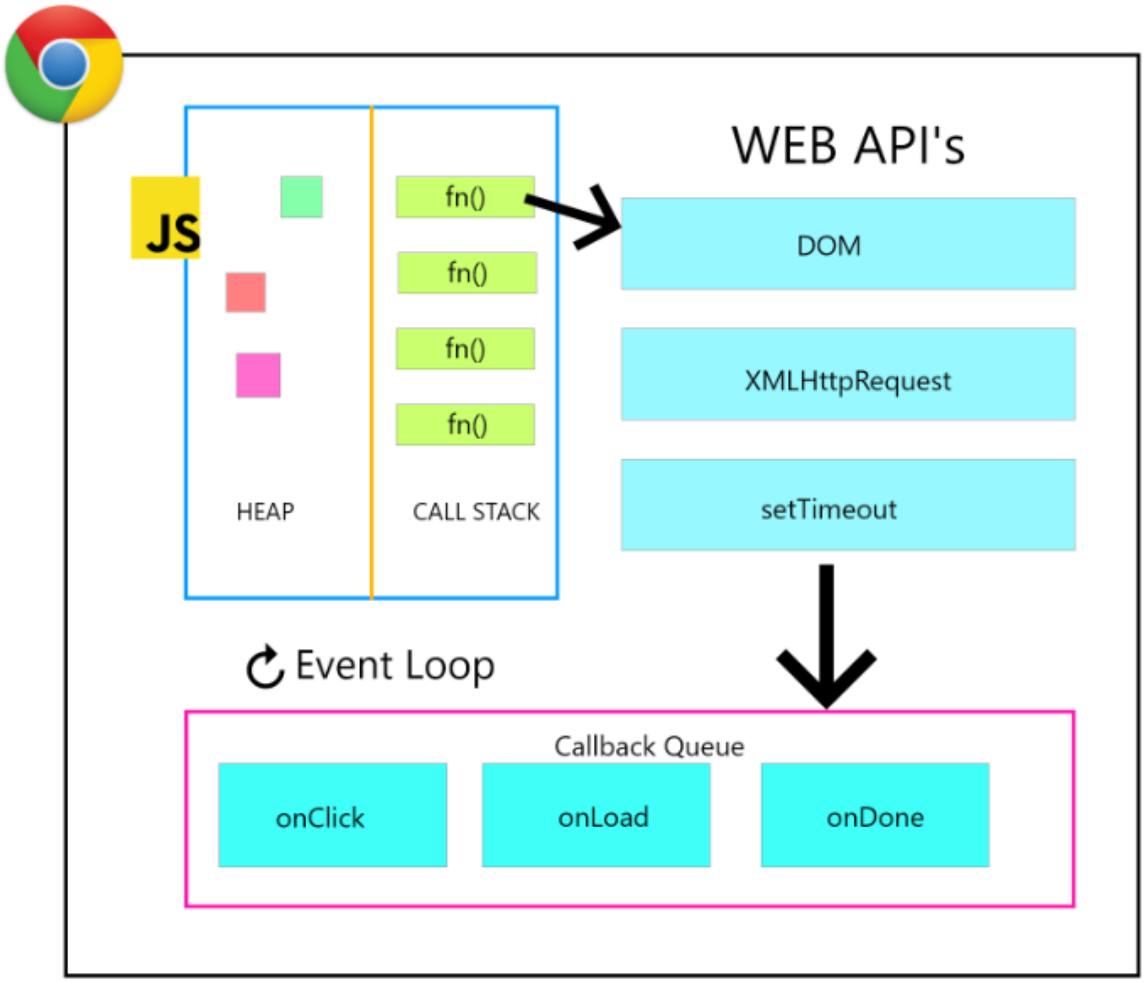
(2) Қолданбалы мәселелер параллельділікке ие болуы керек. Яғни, қосымшаны параллель орындалатын бірнеше ішкі тапсырмаларға бөлуге болады. Қолданбаны бірнеше ішкі тапсырмаларға ыдырату процесі параллельді алгоритмдерді жобалау деп аталады.

(3) Параллель бағдарламалау. Параллельді машинамен қамтамасыз етілген параллельді бағдарламалау ортасында параллельді алгоритм арнайы орындалады, параллель бағдарлама құрастырылады және қолданбалы есептерді параллельде шешу мақсатына жету үшін бағдарлама іске қосылады.

II. JavaScript-дегі параллельді есептеу

2.1 Параллель модельдер және оқиғалар циклі

JavaScript механизмі бір ағында жұмыс істейді. JavaScript тіліндегі уақытты қажет ететін енгізу-шығару (I/O) операциялары, мысалы пернетақта мен тінтуірдің енгізу-шығару оқиғаларын және таймер (setTimeout, setInterval) сияқты оқиғаларын асинхронды операциялар ретінде өңдейді. Бұл асинхронды тапсырмалар орын алған кезде, олар браузердің оқиға тапсырмалары кезегіне қойылады. JavaScript іске қосылған кезде орындау ағыны бос тұрғанда, олар кезектің бірінші келген бірінші шығу принципіне (FIFO) сәйкес бір-бірден орындалады.

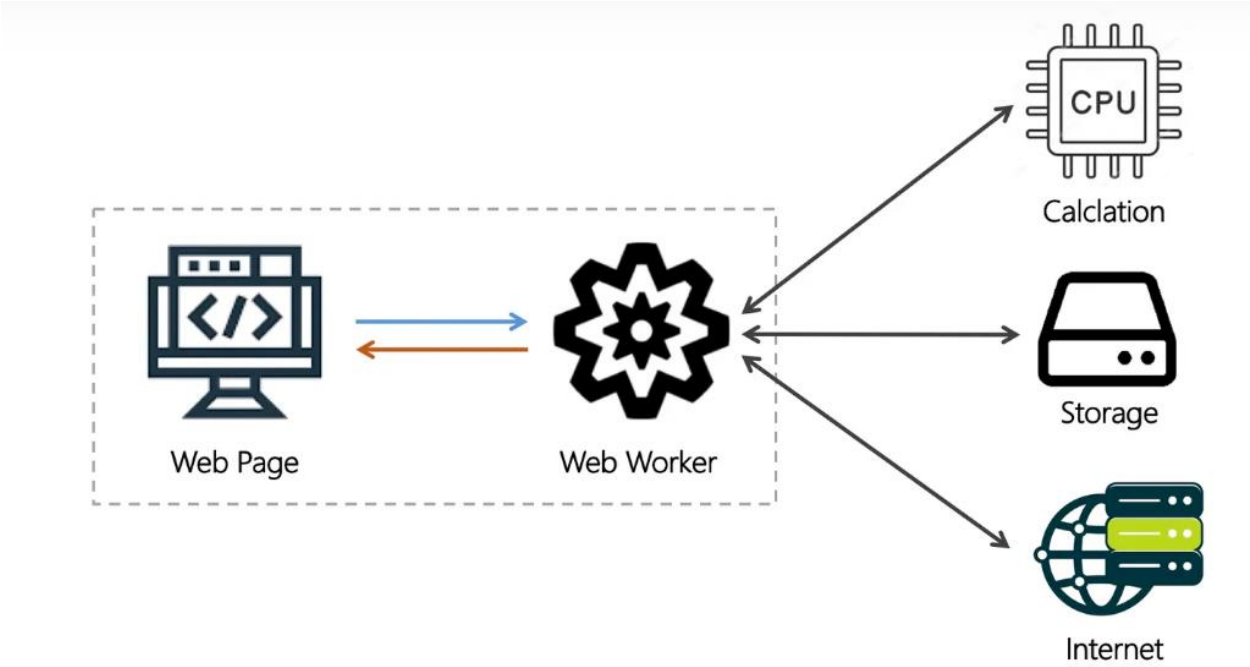


2-сурет. JavaScript Оқиғалар циклі

2.2 Web Worker

Web Workers HTML5 стандартының бөлігі болып табылады, ол JavaScript бағдарламасына негізгі ағыннан басқа ағында жұмыс істеуге мүмкіндік беретін API жиынтығын анықтайды.

Жұмысшы ағындары әзірлеушілерге пайдаланушылар үзбей ұзақ уақыт жұмыс істей алатын фондық бағдарламаларды жазуға, транзакцияларды немесе логиканы орындауға мүмкіндік береді және сонымен бірге пайдаланушыларға дер кезінде жауап беруін қамтамасыз етеді.



4-сурет. WebWorker

The Worker interface spawns real OS-level threads [5].

Web Worker операциялық жүйе деңгейіндегі ағындарды құрады. Worker көп ағыны есептеу өнімділігін тікелей жақсартпайды. Оны жақсартуға болатын-болмайтындығы құрылғының CPU ядроларының санына және ағын стратегиясына байланысты.

Шын мәнінде өнімділікті жақсартуға әкелетін нәрсе - көп ядролы және көп ағынды параллельдік.

Төмендегі суретте кескінді өңдеу демонстрациясында көрсетілгендей, 6 Web Worker жұмыс жасайды, нәтижесінде жалпы кескінді өңдеу уақыты тек негізгі ағынды сериялық өңдеуге қарағанда шамамен 2000 мс жылдамырақ.

5-сурет. Кескінді өңдеу демонстрациясы

Тағы басқа деректер салыстыруларын сайт-тан көруге болады.

Қортынды

Ағымдағы браузерлер Worker-ді жақсы қолдайтынын көруге болады. Worker-ді алдыңғы қатарлы бизнеске қауіпсіз түрде қолдануға болады.

Web Worker-дің көп ағынды мүмкіндігі күрделі фронтальды жобалар үшін стандарт болады деп күтілуде, бұл UI ағынының қатып қалуын азайтуға және компьютердің өнімділігін төмендетуге пайдалы болады. Алайда қазіргі уақытта отандық тәжірибе аз, қоғамдастыққа ғылымды танымал ету және практикалық бөлісу жетіспейді.

Әдебиеттер тізімі

1. [1] Gottlieb, Allan; Almasi, George S. (1989). *Highly parallel computing*. Redwood City, Calif.: Benjamin/Cummings. ISBN 978-0-8053-0177-9.

2. [2] 张林波等，《并行计算导论》，北京：清华大学出版社，2006.6，第3页。

3. [3] Asanovic, Krste et al. (December 18, 2006). "The Landscape of Parallel Computing Research: A View from Berkeley" (PDF). University of California, Berkeley. Technical Report No. UCB/EECS-2006-183. "

4. [4] 张林波等，《并行计算导论》，北京：清华大学出版社，2006.6，第4页。
5. [5] MDN web docs