

EB SOCKET ХАТТАМАСЫ НЕГІЗІНДЕ NODE.JS ОРТАСЫНДА REALTIME ЖӘНЕ АСИНХРОНДЫ АҚПАРАТ АЛМАСУ МҮМКІНДІГІМЕН ҚОСЫМША ӘЗІРЛЕУ

Сансызбай Нұрболат Қанатұлы

nurbolatsnk@gmail.com

Л.Н. Гумилев атындағы ЕҰУ ЕТБҚ мамандығының 1-курс магистранты, Нұр-Сұлтан қ.,
Қазақстан

Ғылыми жетекшісі - ф.-м.ғ.к., профессор Д.Ж. Сатыбалдина

Жіберілетін ақпарат көлемінің геометриялық прогрессиямен өсуі, IoT сенсорларының өндеу құрылғыларына ақпаратты үздіксіз жіберуі, machine learning кластерлеріне актуалды ақпараттың дереу жеткізілуі, сервер – клиент арақатынасында realtime байланыс орнатуды талап етеді.

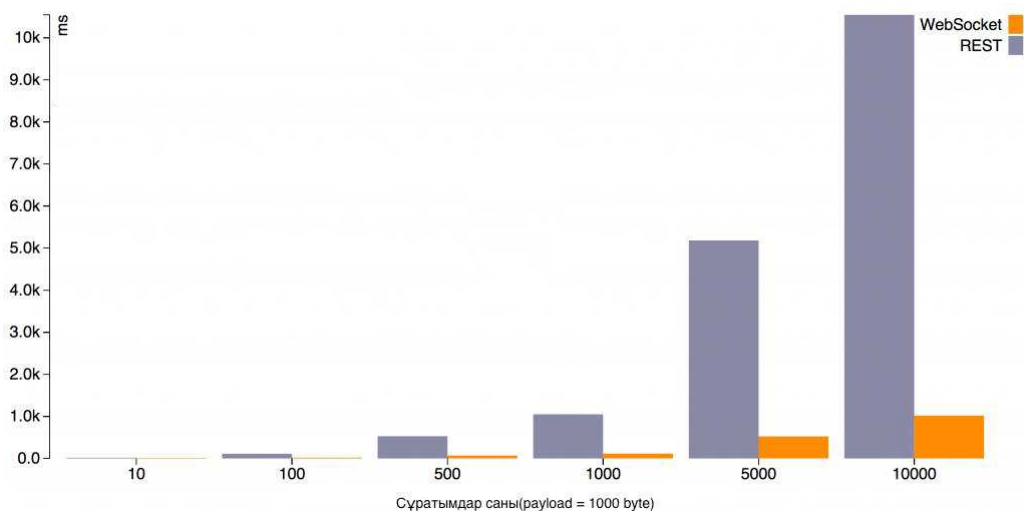
Қазіргі таңда сервер-клиент арасында realtime байланыс орнатудың келесідей әдістері бар:

1. Short pooling: HTTP протоколы негізінде арнайы уақыт аралығында серверге сұратымдар жіберу. Мысалы: Ajax, axios.

2. Long pooling: HTTP протоколы негізінде жасалынған сұратылым бойынша орнатылған байланыстың сервер тоқтатпай, жаңа ақпарат келмегенше үзбейді.

3. WebSocket: WebSocket протоколы негізінде сервер-клиент арасында толыққанды асинхронды ақпарат алмасуға мүмкіндік береді.

Дәстүрлі HTTP хаттамасы жасанды realtime байланыс орнатуға мүмкіндік береді. Алайда, мұндай байланыс, сұратымдардың санының өсуімен қатар сервер құрылғысының қуаттылығын арттыруды талап етеді. Мысал ретінде, тестілеу мақсатында, nodejs ортасында екі web қосымша құрылды: біріншісі – http хаттамасы негізінде сұратымдар қабылдау интерфейсімен (REST), екіншісі – WebSocket хаттамасы негізінде сұратымдар қабылдау интерфейсімен. Бір уақытта бірнеше сұратым жіберу үшін Postman бағдарламасы қолданылды.



Сурет 1 – HTTP және Websocket хаттамаларын сұратым санына байланысты жауап беру уақытын тестілеу

Тестілеу нәтижесіне сәйкес, желінің әсер ету факторларын есептемегенде websocket, http хаттамасымен салыстырғанда ~10 есе эффективтілікті көрсетті. Websocket хаттамасының жоғарыда айтылған артықшылықтарынан басқа келесідей мүмкіндіктерін атап айтуға болады:

- **Bi-directional:** HTTP клиенттің сұратымымен байланыс орнатылатын бір бағытты хаттама. Websocket екібағытты және ақпарат алмасуда үлгі бойынша жіберуді талап етпейді.
- **Full-duplex:** HTTP серверге өздігінен клиентке ақпарат жіберуге мүмкіндік бермейді, ал websocket тәуелсіз сұратылымсыз серверден клиентке ақпарат жіберуді мүмкін етеді.
- **TCP-connection:** HTTP және Websocket хаттамалары tcp байланысы қолданады және OSI моделі бойынша екеуі де Application деңгейінде жұмыс жасайды. Алайда, http жауап жібергеннен кейін байланысты үзеді. Бұл жағдайда әрбір сұратым үшін жаңа TCP байланыс ашады.
- **Жеңіл:** Websocket хаттамасында ақпарат жіберуде алдын ала белгіленген үлгінің(pattern) жоқтығы, тек құнды ақпаратты ғана жіберуге және жіберу форматын бағадарламалаушының өзі таңдауына мүмкіндік береді. Мысалы, HTTP хаттамасының тым көп ақпарат жіберетінің келесі 2-ші суреттегі Header хатынын көруге болады.

```
80 "0.0.0.0"
{"host":"localhost","connection":"keep-alive","upgrade-insecure-requests":"1","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS
X 10_14_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36","accept":"text/html,application/xhtml+xml,
application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3","accept-encoding":"gzip, deflate, br","a
ccept-language":"en-US,en;q=0.9"}
```

Сурет 2 – Серверге жіберілген сұратымның header бөлігі

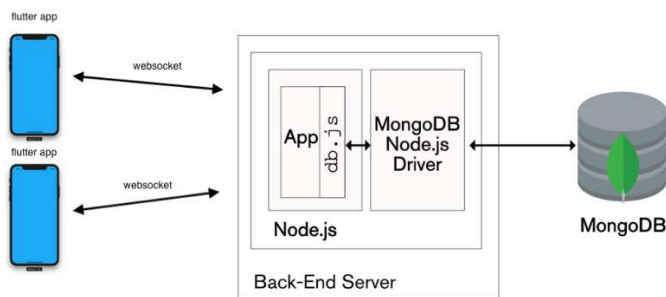
Біздің қосымшамыз, қозғалмалы объектілердің геопозициясын realtime мүмкіндігімен бақылау және дереккорда сақтау. Қосымша құру екі бөліктен тұрады. Біріншісі, Backend-серверлік бөлік, келесідей құралдар мен кітапханаларды қамтиды: nodejs, express-framework, socket.io, mongoDB. Екіншісі, Frontend яғни клиенттік бөлік келесідей: flutter-framework, socket.io-client.

Flutter - бір кодалық базада жазылған қосымшаны ios және android платформаларына орнатуға мүмкіндік беретін, google компаниясымен жасалынған 2018 жылғы framework. Flutter-ды таңдаудағы басты себебіміз, жазылған қосымшаның әмбебаптылығы, виртуалды эмуляторлардың қолжетімділігі және кодты жазу жылдамдығы.

Nodejs – javascript тілінде жазылған кодты серверде орындау мүмкіндігі бар орта. Nodejs асинхронды бағдарламалау стилін ұстанады. Және кодты параллельді емес, тізбектей орындайды. Бұндай қасиеті оперативті жадыны тиімді қолдануға, сұратымдарды жылдам өңдеуге ықпал етеді.

Қосымшаның дерекқоры ретінде NOSQL MongoDB таңдалынды. Сақталынған ақпаратты realtime алуға және дерекқорды горизонтальды өсіруге мүмкіндігі кең болғандықтан біздің қосымшамыз үшін таптырмас шешім.

Қосымшамыздың жалпы архитектурасын 3-ші суреттен көруге болады.



Сурет 3 – Геопозицияны бақылауға арналған қосымшаның архитектурасы

Қорытынды

Қосымшамыз, клиент-сервер арасында websocket хаттамасын қолдану арқылы асинхронды және realtime ақпарат алмаса алады. Мұнымен қоса, MongoDB дерекқорының 4-ші нұсқасы қолданылып, кіріктірілген change streams технологиясының арқасында жүйе толықтай асинхронды және realtime мүмкіндігіне ие болды.

Қолданылған әдебиеттер тізімі

1. Azat Mardan. Practical Node.js, Building Real-World Scalable Web Apps, second edition. Apress. 2018
2. Basarat Ali Syed. Beginning Nodejs. Apress. 2014
3. Socket.io ресми сайты. “Documentation”[Электронды ресурc] <https://socket.io/docs/>
4. Medium ресми сайты. “What are websockets?” [Электронды ресурc]<https://medium.com/front-end-weekly/what-are-websockets-7bf0e2e1af2>