

**“LOAD BALANCER”NING DASTURIY IMKONIYATLARI****G.Rayimjonov***Erkin tadqiqotchi*

**Annotatsiya:** *Maqolada yuk balansi proksi-server sifatida ishlaydigan va tarmoq yoki dastur trafigini bir qator serverlardagi so'nggi nuqtalar bo'ylab tarqatuvchi yechimi sifatida yuk balanslagichlari tirbandlik vaqtida quvvatni taqsimlash va ilovalarning ishonchliligini oshirish uchun ishlatiladigan dastur haqida ma'lumot berilgan.*

**Kalit so'zlar:** *load balancer, Server, API, HTTP, arxitektura, web, Klient server, Microservices*

Kundalik hayotimizda deyarli hammamiz turli xil serviselardan(internetdagi) foydalanamiz. Ma'lumotlar yuklaymiz, ma'lumotlarni ko'chirib olamiz. Qisqacha qilib aytadigan bo'lsak so'rov jo'natib javob olamiz. Bu jarayonlar qanday amalga oshirilayotganini ko'pchiligimiz bilmaymiz. Bitta ma'lumotni olib chiqish jarayoni ortida qanday ishlar olib borilishi haqida tasavvur qilmasligimiz mumkin. Misol uchun google xizmatidan juda ko'p foydalanamiz. Foydalanuvchilar ko'p bo'lishiga qaramay, biz so'ragan ma'lumotlarni tezlik bilan bizga taqdim qiladi. Vaholanki, bir vaqtning o'zida millionlab foydalanuvchi google serviselaridan foydalanadi. Bu so'rovlar bir necha serverlar o'rtasida taqsimlanishni o'z ichiga oladi.

Biz yaratgan dasturimiz yoki veb saytimiz ommalashishni boshlagandan keyin, serverga yuklanish ortib ketadi. Bu esa veb saytni sekin ishlashiga sabab bo'ladi. Katta hajmdagi so'rovlarga tez va aniq javob berishni qo'shimcha serverlar qo'shish(horizontal scaling) va ushbu so'rovlarni server bo'ylab taqsimlash orqali amalga oshirishimiz mumkun. Qaysi so'rovni qaysi serverga yo'naltirishni kim hal qiladi?

Javob : Load balancer

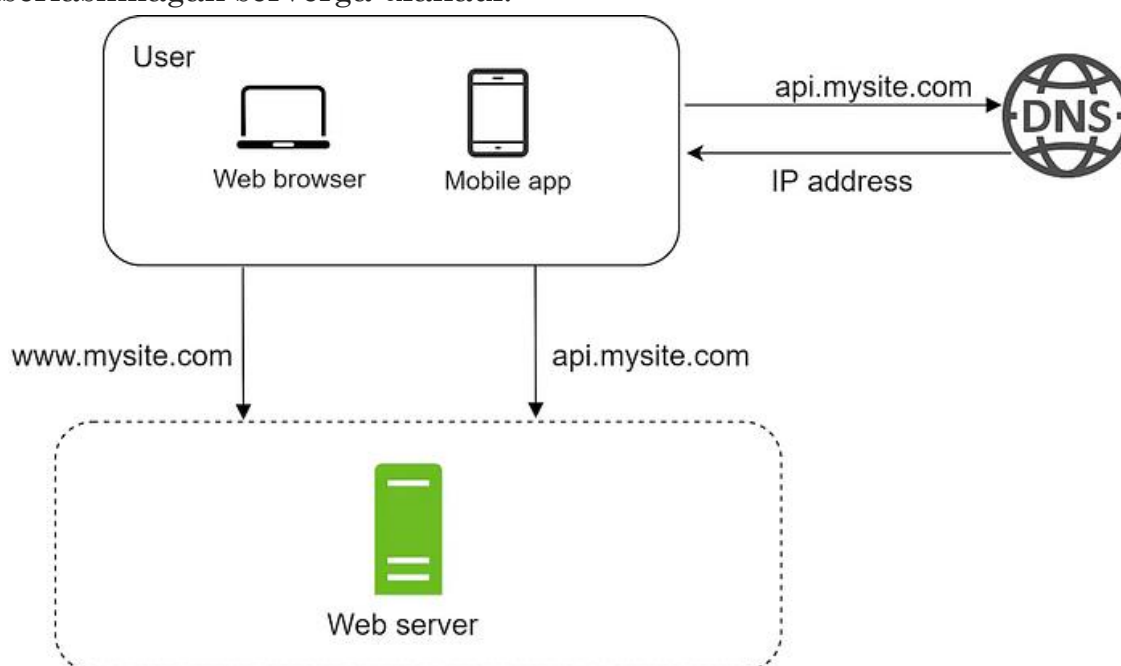
Load balancer nima?

Load balancer nima ekanligini bilish uchun avval trafik taqsimlash nimaligi haqida tasavvur hosil qilib olaylik. Tasavvur qiling ertalabki yo'llardagi tirbandlik(probka), chorrahada turgan DAN(GAI) xodimini ko'z oldingizga keltiring. Inspektor chorxadagi sevetoforlarga qaramay trafikka qarab mashina harakatini yo'naltiradi. Qaysi yo'nalishda trafik ko'p bo'lsa, o'sha yo'nalishni ko'proq ochib kam bo'lgan yo'nalishni kamroq ochib turadi va shu yo'l bilan tirbandliklar uzundan uzoq bo'lishni oldini oladi. Shu bilan inspektor yo'nalishdagi balansni ta'milashga xizmat qiladi. Dasturlashdagi load balancer

ham xuddi shunday ishni bajaradi. So'rovlar ko'payishi bilan so'rovlari kam bo'lgan serverga so'rovlarni yo'naltiradi. Shu usul bilan so'rovlarni "tirbandligini))" oldini oladi.

Load balancer tarmoq orqali kiruvchi trafikni bir nechta manba orqali taqsimlash imkonini beradi. Bu esa so'rovlarni faqat online bo'lgan resurslarga yuborish orqali yuqori aniqlik va ishonchlikni, resurslarni qo'shish yoki ayirish uchun moslashuvchanlikni ta'minlaydi. Load balancer hardware yoki software da ishlaydigan qurilma yoki virtual instance bo'lishi mumkin.

Demak bizda ikki xil ssenariy bo'lishi mumkin. Bular: single server(balanser qilinmagan server) — foydalanuvchi(klient) doim balasertashmagan serverga ulanadi.

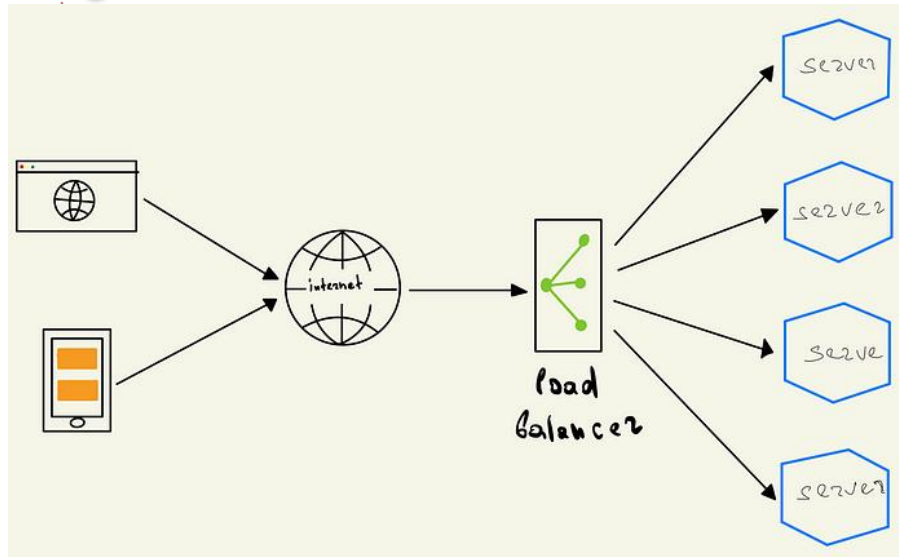


Ushbu modelning ikkita asosiy kamchiligi mavjud:

- **Single Point of Failure:** Agar server o'chib qolsa yoki serverda biror narsa yuz bersa, butun tizim to'xtaydi va ma'lum vaqt davomida foydalanuvchilar tizimdan foydalana olmaydi. Bu — xizmat ko'rsatuvchi tizimlar uchun juda yomon holat.

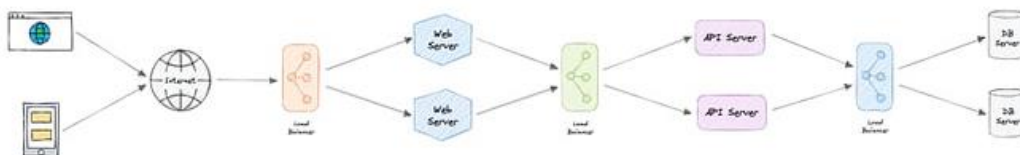
- **Overload Servers:** Har bir serverda bajara oladigan so'rovlar limiti mavjud. Biznes rivojlanishi va so'rovlar soni ko'payishi serverda ortiqcha yuklanishga olib keladi. O'sib borayotgan so'rovlarni sonini hal qilish uchun biz yana bir nechta serverlarni qo'shishimiz kerak bo'ladi.

Demak biz yuqoridagi muamolarni hal qilish uchun veb-serverlar oldiga load balancer joylashtiramiz. Bu esa istalgancha serverlar qo'shishimiz va istalgan sondagi so'rovlarni bajarishga ruxsat berishimiz imkonini beradi. Ba'zi sabablarga ko'ra serverlardan biri offline bo'lsa ham xizmat davom etaveradi.



- Load balancer serverning javob qaytarish vaqtini kamaytiradi va o'tkazish qobiliyatini maksimal darajada orttiradi
- Load balancer so'rovlarni faqat online serverlarga yuborish orqali yuqori aniqlik va ishonchlilikni ta'minlaydi
- Load balancer server so'rovini ko'rib chiqib serverni holatini doim tekshirib turadi
- So'rov yoki talabga qarab Load balancer serverlar sonini qo'shadi yoki olib tashlaydi

Load balancer larni qayerlarga joylashtirishimiz mumkinligini ko'rsatilgan rasm



## LOAD BALANCER TURLARI

Software load balancer hardware balancer ga qaraganda osonroq versiya hisoblanadi. Shuningdek ancha tejamkor va moslashuvchan bo'lib, dasturni yaratish jarayonida birga ishlab chiqiladi. Software load balancer yondashuvi arzonroq va ko'proq moslashuvchanlikni ta'minlaydi. Chunki har qanday standart qurilmaga o'rnatish mumkin (windows, linux va.h). Software balanser ga spring framework dagi load balancer ni misol qilish mumkin.

## HARDWARE

Nomidan ko'rinib turibdiki, biz tarmoq serverlari kalsteri bo'ylab trafikni tarqatish uchun hardware dan foydalanamiz. Ushbu load balancer lar Layer 4–7 routerlar sifatida ham tanilgan va ular HTTP, HTTPS, TCP, va UDP

trafiklarini barcha turlarini boshqarishga qodir. Hardware load balancer lar har bir serverni holatini tekshirib turadi agar serverlardan birortasi to'g'ri javobni bermasa, u darhol serverlarga trafikni yuborishni to'xtatadi. Ushbu balancer larni sotib olish va sozlash qimmatga tushadi.

### LOAD BALANCERLARNING KATEGORIYALARI

#### 1. Layer 4(L4) load balancer

OSI(The Open Systems Interconnection) modeli layer 4 transport layer(TCP/SSL) bo'lib, unda marshrutlash qarorlari qabul qilinadi. Layer 4 load balanceri shunigdek, Network load balanceri deb ham ataladi. Nomidan ko'rinib turibdiki, u trafik uchun marshrutlash qarorlarini qabul qilish uchun tarmoq layer ma'lumotlaridan foydalanadi. U soniyasiga millionlab so'rovlarni boshqarishi mumkin va u TCP/UDP trafigining barcha shakllarini boshqaradi. Ushbu kategoriyadagi balancerlar IP manzillari, switch, router lar bo'ylab trafikni tarqatish orqali foydalanishni maksimal darajada oshiradi.

#### 2. Layer 7 load balancer

Layer 7 load balancer application balancer yoki HTTP(s) balanceri ham deb ataladi. Bu usul load balancer larni eng eski usullaridan biridir. OSI modelida marshutlash qarorlarini bajaradigan amaliy layer (HTTP/HTTPS). Layer 7 load balancingni ta'minlash va server bo'ylab marshrutlash so'rovlarini bajarish uchun HTTP header, cookies, SSL session ID va HTML forma datalaridan foydalanadi.

### LOAD BALANCER ALGORITIMLARI

Qaysi so'rovni qaysi serverga yo'naltirish kerakligini hal qilish uchun bizga load balancer algoritmi kerak bo'ladi. Turli xil tizimlar load balancerlari serverlarni tanlashning turli usullaridan foydalanadi. Kompaniyalar konfiguratsiyaga qarab turli xil load balancer algoritmi texnikasidan foydalanadilar. Ba'zi umumiy load balancer algoritmlari quyida keltirilgan:

1. Round-robin: So'rovlar serverlar bo'ylab ketma-ket yoki rotatsion tarzda taqsimlanadi. Masalan, birinchi so'rov birinchi serverga, ikkinchisi ikkinchi serverga, uchinchi so'rov uchinchi serverga boradi va u barcha so'rovlar uchun davom etadi. Amalga oshirish oson, lekin u serverdagi yukni hisobga olmaydi, shuning uchun serverlardan biri juda ko'p so'rovlarni qabul qilish va ortiqcha yuklanish xavfi mavjud.

2. Weighted Round-robin: Bu round-robin texnikasiga juda o'xshaydi. Yagona farq shundaki, ro'yxatdagi manbalarning har biri vaznli ball bilan ta'minlangan. Olingan ballga qarab so'rov ushbu serverlarga tarqatiladi. Shunday qilib, bu usulda ba'zi serverlar umumiy so'rovning katta qismini oladi.

3. Least Connections: Ushbu usulda so'rov eng kam sonli so'rovlar yoki faol ulanishlarga ega bo'lgan serverga yo'naltiriladi. Buni amalga oshirish uchun

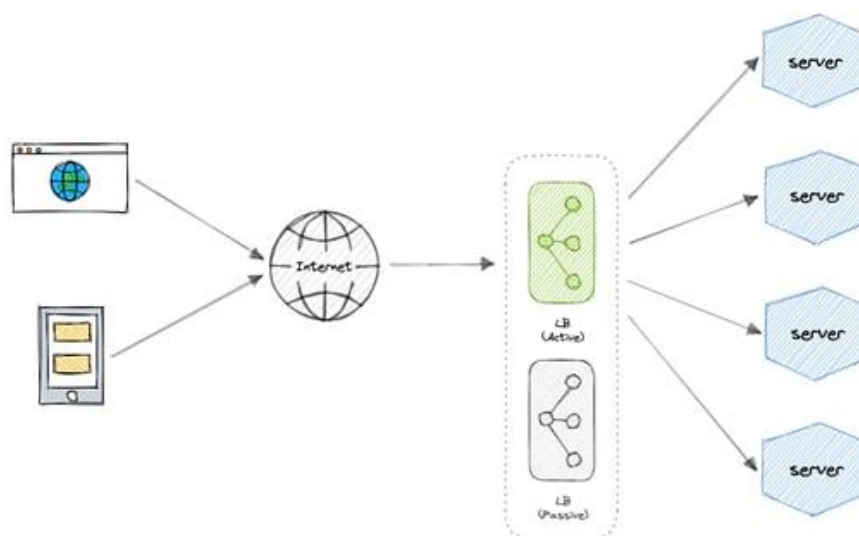
load balancer eng kam ulanishlar soniga ega serverni aniqlash uchun qo'shimcha hisoblashni amalga oshirishi kerak. Bu raund-robin usuli bilan solishtirganda biroz murakkabroq bo'lishi mumkin. Ammo baholash serverdagi joriy yuklamaga asoslanadi. Ushbu algoritm serverlar o'rtasida notekis taqsimlangan trafikda juda ko'p doimiy ulanishlar mavjud bo'lganda foydalidir.

4. Least Response Time: Ushbu uslub Least ulanish usuliga qaraganda ancha murakkab. Ushbu usulda so'rov eng kam faol ulanishlar va eng kam o'rtacha javob vaqti bilan serverga yo'naltiriladi.

5. Hashing: Bu usulda mijozning IP manzili asosida so'rov serverga yuboriladi. Mijozning IP manzili va qabul qiluvchi hisoblash instance kriptografik algoritm bilan hisoblanadi.

Qo'shimcha(ehityojdan ortiqcha) load balancerlar

Tabiiy savol tug'ilishi mumkin: "load balancerlar o'chib qolsa nima bo'ladi?". Bu juda o'rinli savol. Agar load balancer bitta bo'lsa va u o'chib qolsa tizimga ulanish imkoni bo'lmay qolishi mumkin. Shuning uchun ham klaster rejimida ikkita yoki N ta load balancerlardan foydalanishimiz mumkin.



System design bo'yicha interview paytida load balancerdan qanday foydalanish kerak?

System design bo'yicha intervyuda sizdan qandaydir masshtablilik savoli so'raladi, unda siz load balancer lar trafikni qanday taqsimlashda yordam berishini va ilovangizdagi xizmatlarning kengayishi va mavjudligini qanday ta'minlashini tushuntirishingiz kerak bo'ladi. Ushbu maqoladan yodda tutishingiz kerak bo'lgan umumiy tushuncha

- Load balancer ma'lumotlarning ishlashi va o'tkazuvchanligini yaxshilaydigan elastik scalability ni ta'minlaydi. Tizimning mavjudligini

ta'minlash uchun ma'lumotlarning ko'p nusxalarini saqlashga imkon beradi. Agar server ishlamay qolsa yoki butunlay yaroqsiz holga kelib qolsa, xizmatlarni tiklash uchun sizda zaxira nusxasi bo'ladi.

- Load balancer har qanday dastur ichiga joylashtirilishi mumkin.
- Ko'pgina kompaniyalar o'z tizimidagi turli o'lchov nuqtalariga qarab load balancerlarni amalga oshirish uchun apparat va dasturiy ta'minotdan foydalanadilar.

### REFERENCES:

1. <https://devops-journey.uz/guides/web-server/load-balancing>
2. Qulmatov Qurvonali Zokirali o'g'li, 2 Olimjonov O. O. o'g'li . (2023). Mikroservislar va klient-server arxitekturalarining tahlili. <https://doi.org/10.5281/zenodo.7860908>
3. Building Microservices: Designing Fine-Grained Systems by Sam Newman Paperback [64-108 p].
4. Monolith to Microservices: Refactoring Approaches compared: Transforming Applications to cloud-ready Software Architectures by Jonas Fritzsche /Apr 24, 2018 [64-80 p]
5. Building Event-Driven Microservices: Leveraging Organizational Data at Scale 1st Edition by Adam Bellemare 2020 [189-192 p]
6. [https://docs.webmethods.io/api/10.11.0/webmethods\\_api\\_cloud\\_\\_api\\_gateway\\_user\\_s\\_guide/chapter1/#gsc.tab=0](https://docs.webmethods.io/api/10.11.0/webmethods_api_cloud__api_gateway_user_s_guide/chapter1/#gsc.tab=0)