

**DASTURIY TA'MINOT TIZIMLARINI LOYIHALASHDA MIJOS-SERVER
MODELINING O'RNI**

Narziyev Nosir Baxshilloevich

TATU ATDT kafedrası katta o'qituvchisi narziev@tuit.uz

Ibroximov Ibratjon Iqboljon o'g'li

TATU DIF talabasi ibratjonibroximov@gmail.com

Annotatsiya: *Bu maqola dasturiy ta'minot tizimlarini loyihalashda BPMN ning o'rni, uning xususiyatlari va afzalliklari hamda bazi namunalar bilan yoritilgan. Bu maqolada dasturiy taminot tizimlarini loyihalshda BPMN dan foydalanishning yangi qirralri ochib berilgan.*

KIRISH

Mijoz-server arxitekturasi dasturiy ta'minot injiniringida keng qo'llaniladigan dizayn namunasi bo'lib, u turli tizim komponentlari o'rtasida samarali aloqa va tashvishlarni ajratish imkonini beradi. U tizimni ikkita asosiy qismga ajratadi: mijoz va server.

Mijoz-server arxitekturasi nima?

Mijoz-server arxitekturasi ikki kishi o'rtasidagi suhbatga o'xshaydi. Tasavvur qiling-a, pizza buyurtma qiling. Siz (mijoz) pitsa joyiga (serverga) qanday pitsa istayotganingizni ayting va ular uni tayyorlab, sizga yetkazib berishadi. Dasturiy ta'minot nuqtai nazaridan, mijoz foydalanuvchi bilan o'zaro aloqada bo'lib, so'rovlarni serverga yuboradi, u ularni qayta ishlaydi va kerakli ma'lumot yoki natijalarni beradi. Ular HTTP yoki TCP/IP kabi protokollar yordamida tarmoq orqali muloqot qilishadi.

Mijoz-server arxitekturasi turli xil ilovalarda, veb-sahifalardan tortib onlayn xaridlargacha qo'llaniladi. U masshtablilik, modullik, xavfsizlik va ishlashni optimallashtirish afzalliklarini taklif etadi. Mijoz foydalanuvchi interfeysiga e'tibor qaratadi, server esa biznes mantig'i, ma'lumotlarni qayta ishlash va saqlash bilan shug'ullanadi.



Mijoz-server arxitekturasing afzalliklari

1. **Masshtablilik** : Dasturiy ta'minotning muhim xususiyati bu tizimning ko'proq serverlarni qo'shish orqali ortib borayotgan yukni boshqarish qobiliyatidir. Bu tizimning ko'plab mijozlarni ishdan chiqishi yoki sekinlashtirmasdan samarali ishlashini ta'minlaydi. Tizimga qo'shimcha serverlar qo'shilishi yukni taqsimlashi va yanada ko'proq mijozlarni oson boshqarishi mumkin.

2. **Modullik** : Mijoz va server o'rtasidagi tashvishlarni ajratish muhim dasturiy xususiyatdir. Bu ishlab chiqish, sinov va texnik xizmat ko'rsatishni ancha boshqariladigan qiladi. Mijoz va server o'rtasidagi tashvishlarni ajratib, tizimning boshqa qismlariga ta'sir qilmasdan tizimga o'zgartirishlar kiritish mumkin. Bu tizimdagi xatolarni topish va tuzatishni ham osonlashtiradi.

3. **Xavfsizlik** : Serverda ma'lumotlar va mantiqni markazlashtirish muhim dasturiy xususiyatdir. Bu tizimga boshqariladigan kirish imkonini beradi, bu esa ruxsatsiz kirish xavfini kamaytiradi. Serverda ma'lumotlar va mantiqni markazlashtirish ma'lumotlarni boshqarish va himoya qilishni osonlashtiradi. Bu shuningdek, maxfiy ma'lumotlarning ruxsatsiz kirishdan himoyalanganligini ta'minlaydi.

4. **Ishlash** : resurs talab qiladigan vazifalarni serverga yuklash muhim dasturiy xususiyatdir. Bu mijozga sezgir foydalanuvchi interfeysini taqdim etishga e'tibor qaratish imkonini beradi. Mijoz resursni ko'p talab qiladigan vazifalarni serverga yuklash orqali tizim og'ir yuk ostida bo'lsa ham javob berishi mumkin. Bu foydalanuvchi tajribasini yaxshilaydi va tizim og'ir yukga qaramay samarali bo'lishini ta'minlaydi.

Mijoz-server arxitekturasining salbiy tomonlari

1. **Bitta nosozlik nuqtasi** : Mijoz-server arxitekturasini bilan bog'liq mumkin bo'lgan muammolardan biri shundaki, serverning ishlamay qolishi butun tizimni buzishi va uni mijozlarga kirishi mumkin emas. Buni zaxira serverlari yoki o'zgartirish tizimlari kabi ortiqcha choralarni qo'llash orqali kamaytirish mumkin.

2. **Tarmoqqa bog'liqlik** : Mijoz-server arxitekturasini bilan bog'liq yana bir qiyinchilik tarmoq aloqasiga tayanish bo'lib, unumdorlik va sezgirlikka ta'sir qilishi mumkin. Biroq, bu tarmoq protokollarini optimallashtirish va keshlash mexanizmlarini amalga oshirish kabi turli xil texnikalar orqali hal qilinishi mumkin.

3. **Murakkablikning ortishi** : Mijoz-server arxitekturasini amalga oshirish qo'shimcha murakkablikni keltirib chiqarishi mumkin, bu aloqa protokollarini puxta ishlab chiqish va boshqarishni, parallelizm va ma'lumotlarning izchilligini talab qiladi. Biroq, bu murakkablik xavfsizlikni oshirish va resurslardan yanada samarali foydalanish kabi afzalliklarni ham berishi mumkin.

4. **Masshtablilik muammolari** : Yukni bir nechta serverlar bo'ylab teng ravishda taqsimlash yukni muvozanatlash usullarini va tizimni ehtiyotkorlik bilan loyihalashni talab qiladi. Biroq, bu yaxshilangan nosozliklarga chidamlilik va yuqori mavjudlik kabi afzalliklarni ham berishi mumkin.

5. **Cheklangan oflayn funktsionallik** : Mijoz-server arxitekturasining bir cheklovi shundaki, u odatda tarmoq ulanishini talab qiladi, bu esa oflayn yoki past

ulanish stsenariylarida funktsionallikni cheklaydi. Biroq, bu mahalliy keshlash yoki oflayn ma'lumotlarni sinxronlash mexanizmlari kabi turli xil texnikalar orqali hal qilinishi mumkin.

Mijoz-server arxitekturasini qachon tanlash kerak

Mijoz-server arxitekturasi foydali bo'lishi mumkin bo'lgan quyidagi stsenariylarni ko'rib chiqing:

1. Markazlashtirilgan ma'lumotlar va mantiq : Mijoz-server arxitekturasi markazlashtirilgan ma'lumotlarni saqlash va qayta ishlash zarur bo'lganda yechim berishi mumkin. Ma'lumotlarni saqlash va boshqarish uchun alohida qurilmalarga tayanish o'rniga, markaziy server barcha ma'lumotlarni saqlashi va qayta ishlash vazifalarini bajarishi mumkin.

2. Mijozlarga qo'yiladigan talablar : Ko'p mijozlar bilan ishlashni talab qiladigan yoki kelajakda o'sishni kutadigan tizimlar uchun mijoz-server arxitekturasi yukni boshqarishga yordam beradi. Tizim bir nechta mijozlar so'rovlarini bajara oladigan markaziy serverga ega bo'lish orqali yanada samaraliroq miqyoslashi mumkin.

3. Xavfsizlik va kirishni boshqarish : Mijoz-server arxitekturasi nozik ma'lumotlar bilan ishlashda yoki qattiq kirish nazoratini talab qilganda yaxshiroq xavfsizlikni ta'minlaydi. Markaziy serverda ma'lumotlarni saqlash ruxsatsiz kirishdan yaxshiroq himoyalangan bo'lishi mumkin. Kirish nazorati markaziy serverda ham samaraliroq amalga oshirilishi mumkin, bu xavfsizlik buzilishining oldini olishga yordam beradi.

4. Modullik va tashvishlarni ajratish : mijoz-server arxitekturasi foydalanuvchi interfeysini biznes mantig'i va ma'lumotlarni qayta ishlashdan ajratishi mumkin. Server ma'lumotlarni qayta ishlash va biznes mantig'ini bajarishi orqali mijozni soddalashtirish va taqdimot mantig'iga yo'naltirish mumkin.

5. Ishlashni optimallashtirish : Resurs talab qiladigan vazifalar uchun mijoz-server arxitekturasi serverga ularni yanada samaraliroq boshqarish imkonini beradi. Mijoz resursni ko'p talab qiladigan vazifalarni serverga yuklash orqali sezgir va tezkor bo'lib qolishi mumkin.

6. Bir nechta mijoz turlari : veb-brauzerlar, mobil ilovalar va ish stoli ilovalari kabi turli xil mijoz turlarini qo'llab-quvvatlaganda, mijoz-server arxitekturasi yagona yechimni taqdim etishi mumkin. Bir nechta mijozlar so'rovlarini ko'rib chiqa oladigan markaziy serverga ega bo'lgan holda tizimni saqlash va yangilash osonroq bo'ladi.

7. Ma'lumotlarning izchilligi va yaxlitligi : Ma'lumotlarning mustahkamligi va yaxlitligi zarur bo'lganda, mijoz-server arxitekturasi yaxshiroq boshqaruvni ta'minlaydi. Ma'lumotlarni boshqaradigan markaziy server barcha ma'lumotlarning izchil va to'g'riligini ta'minlashi mumkin.

8. Resurs almashish : Fayllar, ma'lumotlar bazalari yoki apparat qurilmalari kabi resurslarni almashishi kerak bo'lgan tizimlar uchun mijoz-server arxitekturasi resurslarni almashishni boshqarish usulini taqdim etishi mumkin. Resurslarni

boshqaradigan markaziy serverga ega bo'lgan holda, u resurslarning to'g'ri va samarali taqsimlanishini ta'minlaydi.

9. Birgalikda ishlash : Bir nechta foydalanuvchilar real vaqtda ma'lumot almashishi va o'zaro ishlashi kerak bo'lganda, mijoz-server arxitekturasi hamkorlikni boshqarish usulini taqdim etishi mumkin. Ma'lumotlarni boshqaradigan markaziy server barcha foydalanuvchilarning bir xil dolzarb ma'lumotlar bilan ishlashini ta'minlashi mumkin.

10. Kelajakda kengaytirilishi : Tizim rivojlanishi va yangi funksiyalar yoki funksiyalarni o'z ichiga olishi kutilayotganda, mijoz-server arxitekturasi o'zgarishlarni boshqarish usulini taqdim etishi mumkin. Ma'lumotlar va biznes mantiqini boshqaradigan markaziy serverga ega bo'lish orqali tizimga yangi xususiyatlar va funktsionallikni qo'shish osonroq bo'lishi mumkin.

Xulosa

Xulosa qilib aytadigan bo'lsak, mijoz-server arxitekturasi tizimingizning maqsad va vazifalariga mos kelishini aniqlash uchun tizim talablarini sinchkovlik bilan tahlil qilish, miqyoslilik, xavfsizlik, modullik va ishlashni optimallashtirish omillarini hisobga olish alohida kasb etadi.

FOYDALANILGAN ADABIYOTLAR:

1. Aruldoss, M.; Lakshmi, M.T.; Venkatesan, V.P. A survey on multi criteria decision making methods and its applications. Am. J. Inf. Syst. 2013, 1, 31–43. [Google Scholar]

2. Velasquez, M.; Hester, P.T. An analysis of multi-criteria decision making methods. Int. J. Oper. 2013, 10, 56–66. [Google Scholar]

3. Hajduk, S. Multi-Criteria Analysis in the Decision-Making Approach for the Linear Ordering of Urban Transport Based on TOPSIS Technique. Energies 2021, 15, 274. [Google Scholar] [CrossRef]