GETEROGEN HISOBLASH TIZIMLARIDA EXECUTION-FEEDBACK ASOSIDAGI INTELLEKTUAL YUKLAMA BALANSLASH MODELI

Ergashev Shaxboz Toshtemir o'g'li

Muhammad al-Xorazmiy nomidagi TATU Assistant o'qituvchisi
Denov tadbirkorlik va pedagogika instituti tayanch doktoranti
Email:toshtemirovich144@gmail.com

K E Y W O R D S

Bulutli hisoblash, yuklamani balanslash, geterogen muhit, so'rovlarni tasniflash, EMA stabilizatsiyasi, CloudSim Plus

A B S T R A C T

Ushbu maqolada geterogen bulutli hisoblash muhitlarida resurslarni samarali taqsimlash va server yuklamalarini optimallashtirishga qaratilgan Adaptive Request-Aware Feedback-Based Load Balancing modeli taklif etilgan. Zamonaviy yuklama balanslash algoritmlarining asosiy kamchiligi sifatida kiruvchi so'rovlarning individual murakkabligini hamda request bajarilgandan keyingi real ijro natijalarini yetarli darajada hisobga olmasligi ko'rsatib o'tilgan. Taklif etilayotgan yondashuvda so'rovlar light, medium va heavy toifalarga ajratilib, ularning hisoblash og'irligi serverlarning joriy holati bilan birgalikda tahlil qilinadi. Modelda har bir request bajarilgandan keyin hosil bo'ladigan CPU yuklanishi va javob qaytarish vaqti feedback mexanizmi orqali yig'ilib, keyingi taqsimlash qarorlarini adaptiv ravishda yaxshilash uchun foydalaniladi. Tizimdagi beqaror tebranishlarni kamaytirish maqsadida Exponential Moving Average (EMA) asosidagi stabilizatsiya mexanizmi qo'llanilgan. Tadqiqot CloudSim Plus simulyatsion muhiti asosida amalga oshirildi. Simulyatsiyada bir nechta virtual mashinalardan iborat geterogen server muhiti yaratilib, turli intensivlikdagi workloadlar yordamida tajribalar o'tkazildi. Taklif etilgan algoritmning samaradorligini baholash maqsadida u an'anaviy Round Robin (RR) hamda Least Connections (LC) algoritmlari bilan taqqoslandi. Eksperimental natijalar adaptive feedback asosidagi yondashuvning o'rtacha javob qaytarish vaqtini kamaytirish va resurslardan foydalanish samaradorligini oshirishda yuqori natija ko'rsatganini tasdiqladi. Xususan, taklif etilgan model Round Robin algoritmiga nisbatan 16.7% gacha yaxshiroq javob qaytarish vaqtini ta'minladi hamda serverlar o'rtasidagi yuklama nomutanosibligini sezilarli darajada kamaytirdi. Olingan natijalar request-aware va execution-feedback asosidagi adaptiv yondashuvlarning dinamik bulutli tizimlarda resurslardan foydalanish samaradorligini oshirish uchun istiqbolli yechim ekanligini ko'rsatadi.

Kirish

Zamonaviy bulutli hisoblash va taqsimlangan axborot tizimlarining jadal rivojlanishi natijasida foydalanuvchi so'rovlari soni, hajmi va hisoblash murakkabligi keskin ortib bormoqda. Mikroservis arxitekturalari, sun'iy intellekt servislarining keng qo'llanilishi, real vaqt rejimidagi multimedia oqimlari hamda katta hajmdagi ma'lumotlarni qayta ishlash jarayonlari server infratuzilmalariga yuqori dinamik yuklama hosil qilmoqda. Bunday sharoitda hisoblash resurslarini optimal taqsimlash va yuklamani samarali balanslash tizimning javob qaytarish vaqti, xizmat ko'rsatish sifati (QoS), resurslardan foydalanish samaradorligi va umumiy barqarorligini belgilovchi asosiy omillardan biri hisoblanadi [1].

Bulutli hisoblash muhitlarida qo'llaniladigan an'anaviy yuklama balanslash algoritmlari, jumladan Round Robin (RR), Weighted Round Robin (WRR), Least Connections (LC), Min-Min va Max-Min kabi yondashuvlar uzoq yillardan buyon amaliy tizimlarda qo'llanib kelinmoqda. Ushbu algoritmlar soddaligi va tez ishlashi bilan ajralib tursa-da, ular asosan statik yoki yarim dinamik mezonlarga asoslanadi. Mazkur usullar kiruvchi so'rovlarning individual murakkabligini, bajarilish xarakteristikasini hamda request bajarilgandan keyin serverda yuzaga keladigan real yuklama holatini yetarli darajada hisobga

(cc) BY

olmaydi. Natijada geterogen server muhitlarida ayrim hisoblash tugunlarining ortiqcha yuklanishi, javob qaytarish vaqtining ortishi va resurslardan nomutanosib foydalanish holatlari yuzaga keladi [2].

So‘nggi yillarda fuzzy logic, machine learning va deep reinforcement learning asosidagi aqlli yuklama balanslash yondashuvlari taklif qilinmoqda. Ushbu usullar tizim holatini dinamik baholash va optimal qaror qabul qilish imkoniyatini oshirsa-da, mavjud ishlarda bir qator muammolar saqlanib qolmoqda. Jumladan, ko‘plab tadqiqotlarda qaror qabul qilish faqat serverning joriy holatiga asoslanib, request bajarilgandan keyingi real ijro natijalaridan adaptiv ravishda foydalanish mexanizmlari yetarli darajada o‘rganilmagan. Bundan tashqari, ayrim yondashuvlar sintetik workloadlar asosida baholanganligi sababli ularning real muhitdagi samaradorligi va barqarorligi yetarlicha isbotlanmagan [3].

Feedback asosidagi tizimlarda yana bir muhim muammo sifatida “oscillation” yoki “herd effect” hodisasi kuzatiladi. Bunda ma’lum bir serverning bo‘shligi aniqlangach, requestlarning asosiy qismi aynan shu tugunga yo‘naltiriladi va natijada u qisqa vaqt ichida ortiqcha yuklanadi. Bu esa tizimda barqarorlikning pasayishiga va resurslardan foydalanish samaradorligining kamayishiga olib keladi. Shu sababli zamonaviy yuklama balanslash tizimlari nafaqat request va server holatini tahlil qilishi, balki feedback asosida o‘z qarorlarini adaptiv ravishda yangilab borishi ham talab etiladi [4].

Mazkur tadqiqotda geterogen bulutli hisoblash muhitlari uchun Adaptive Request-Aware Feedback-Based Load Balancing modeli taklif etiladi. Taklif etilgan yondashuvda kiruvchi requestlarning turi, hajmi va hisoblash og‘irligi serverlarning real vaqt rejimidagi holati bilan birgalikda tahlil qilinadi. Modelning asosiy yangiligi shundan iboratki, request bajarilgandan keyin hosil bo‘ladigan real CPU yuklanishi va javob qaytarish vaqti feedback mexanizmi orqali yig‘ilib, keyingi taqsimlash qarorlarini adaptiv ravishda yaxshilash uchun foydalaniladi. Tizimdagi tebranishlarni kamaytirish va barqarorlikni oshirish maqsadida Exponential Moving Average (EMA) asosidagi stabilizatsiya mexanizmi qo‘llanilgan [5].

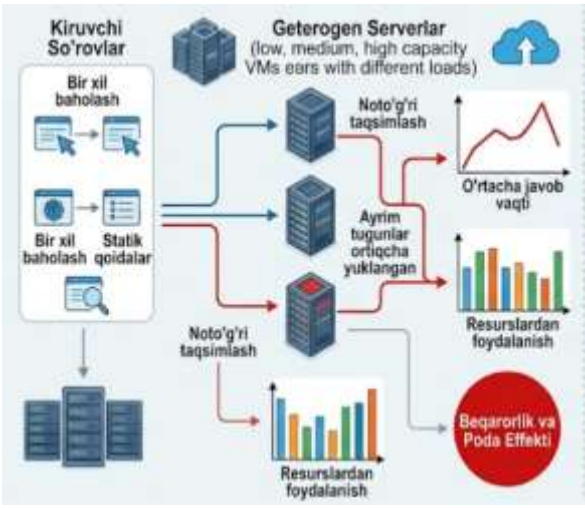
Ushbu tadqiqotning asosiy ilmiy yangiliklari quyidagilardan iborat:

Birinchidan, kiruvchi so‘rovlarni hisoblash og‘irligiga ko‘ra light, medium va heavy toifalarga dinamik ajratishga asoslangan Request-Aware Classification mexanizmi taklif etildi. Mazkur yondashuv requestlarning individual hisoblash murakkabligini hisobga olish imkonini beradi.

Ikkinchidan, faqat serverning joriy holatiga emas, balki request bajarilgandan keyingi real CPU yuklanishi va javob qaytarish vaqti ko‘rsatkichlariga asoslangan Execution-Based Feedback Loop mexanizmi ishlab chiqildi. Ushbu yondashuv keyingi taqsimlash qarorlarini adaptiv ravishda yaxshilash imkonini beradi [6].

Uchinchidan, feedback tizimlarida kuzatiladigan beqarorlik va “poda effekti” (herd effect) muammosini kamaytirish maqsadida Exponential Moving Average (EMA) asosidagi stabilizatsiya mexanizmi joriy etildi. Ushbu mexanizm serverlar o‘rtasidagi yuklama tebranishlarini yumshatishga xizmat qiladi [7].

To‘rtinchidan, taklif etilgan model CloudSim Plus simulyatsion muhiti asosida turli intensivlikdagi workloadlar ostida sinovdan o‘tkazildi hamda uning samaradorligi Round Robin (RR) va Least Connections (LC) algoritmlari bilan taqqoslash orqali baholandi.



1-rasm.An’anaviy yondashuvlar

1-rasmda kiruvchi so‘rovlar (Light, Medium, Heavy bo‘lishiga qaramay) tizim tomonidan ‘bir xil’ deb baholanadi va geterogen (turli quvvatdagi) serverlarga ketma-ket (Round Robin)

(cc) BY

yoki ulanishlar soniga qarab (Least Connections) noto‘g‘ri taqsimlanadi.

Rasmda ko‘rinib turibdiki, ayrim tugunlar haddan tashqari yuklanib ketadi (qizil server), o‘rtacha javob berish vaqti cho‘ziladi va resurslardan foydalanish balansi buziladi. Pastdagi qizil doira matnda aytilgan ‘Beqarorlik va Poda Effekti’ (grafikdagi keskin sakrashlar) muammosini yaqqol ko‘rsatadi.

Taklif etilgan modelning samaradorligi CloudSim Plus simulyatsion muhiti asosida tadqiq qilindi. Tajribalarda turli intensivlikdagi workloadlar va geterogen virtual mashinalardan tashkil topgan server muhiti yaratildi. Algoritmning ishlash samaradorligi Round Robin (RR) hamda Least Connections (LC) algoritmlari bilan taqqoslandi. Eksperimental natijalar taklif etilgan adaptiv yondashuvning o‘rtacha javob qaytarish vaqtini kamaytirish, resurslardan foydalanish samaradorligini oshirish hamda serverlar o‘rtasidagi yuklama nomutanosibligini pasaytirishda yuqori natija ko‘rsatishini tasdiqladi [8].

Adabiyotlar tahlili.

Bulutli hisoblash muhitlarida yuklama balanslash muammosi ko‘plab tadqiqotchilar tomonidan o‘rganilgan bo‘lib, ushbu yo‘nalishdagi dastlabki ishlarda asosan statik algoritmlarga asoslangan yondashuvlar qo‘llanilgan. E. Casalicchio va hamkorlari tomonidan olib borilgan tadqiqotlarda Round Robin (RR) va Weighted Round Robin (WRR) algoritmlarining bulutli muhitdagi samaradorligi tahlil qilingan. Round Robin algoritmi requestlarni serverlarga ketma-ket taqsimlash tamoyiliga asoslanib, implementatsiya jihatidan sodda va tez ishlashi bilan ajralib turadi. Biroq ushbu algoritm serverlarning real yuklanishi yoki requestlarning hisoblash murakkabligini hisobga olmaydi. Natijada geterogen server muhitlarida resurslardan nomutanosib foydalanish holatlari yuzaga keladi [9].

Weighted Round Robin algoritmida serverlarning hisoblash quvvatiga mos ravishda og‘irlik koeffitsientlari belgilanadi. H. Liu va hamkorlari tomonidan o‘tkazilgan tadqiqotlarda WRR algoritmi RR ga nisbatan samaraliroq natija ko‘rsatishi qayd etilgan. Shunga qaramasdan, mazkur usul ham requestlarning dinamik xususiyatlari va real vaqt rejimidagi server holatini to‘liq hisobga olmaydi. Serverning avvalgi ishlash tarixi yoki request bajarilgandan keyingi holati ushbu algoritmda inobatga olinmaydi [10].

Dinamik yuklama balanslash usullaridan biri sifatida Least Connections (LC) algoritmi ko‘plab amaliy tizimlarda qo‘llanilgan. M. Randles va D. Lamb tomonidan olib borilgan ishlarda LC algoritmi aktiv ulanishlar soniga qarab server tanlash mexanizmini qo‘llashi sababli RR algoritmiga nisbatan moslashuvchanroq ekanligi ko‘rsatilgan. Biroq ushbu yondashuvda ham requestning real hisoblash murakkabligi, CPU bandligi yoki response time kabi parametrlar yetarli darajada hisobga olinmaydi [11].

Task scheduling va resurslarni optimal taqsimlash masalalarida Min-Min va Max-Min algoritmlariga asoslangan heuristik yondashuvlar ham keng tadqiq qilingan. R. Buyya va hamkorlari tomonidan CloudSim platformasi asosida o‘tkazilgan tadqiqotlarda ushbu algoritmlar task execution time ni qisqartirishda ma’lum samaradorlik ko‘rsatgan. Shunga qaramasdan, heuristik yondashuvlar asosan statik workload modellari asosida ishlaydi va dinamik o‘zgaruvchan muhitlarga tez moslasha olmaydi [12].

So‘nggi yillarda fuzzy logic asosidagi aqlli yuklama balanslash tizimlariga qiziqish ortdi. S. Wang va hamkorlari tomonidan taklif etilgan fuzzy-based load balancing modelida CPU yuklanishi va xotira bandligi asosida server tanlash mexanizmi ishlab chiqilgan. Ushbu yondashuv noaniq va dinamik muhitlarda an’anaviy algoritmlarga nisbatan samaraliroq natija ko‘rsatgan. Biroq ko‘plab fuzzy logic asosidagi ishlarda requestlarning individual hisoblash og‘irligi yetarli darajada baholanmagan [13].

Machine learning va deep reinforcement learning asosidagi zamonaviy tadqiqotlarda tizimning avvalgi holatlaridan o‘rganish orqali optimal server tanlash mexanizmlari ishlab chiqildi. X. Xu va hamkorlari tomonidan taklif etilgan deep reinforcement learning asosidagi model dinamik workloadlar ostida yuqori samaradorlik ko‘rsatgan. Biroq ushbu usullar yuqori hisoblash xarajatlari, murakkab trening jarayoni va amaliy tizimlarga integratsiya qilishdagi qiyinchiliklar bilan tavsiflanadi.

(cc) BY

Bundan tashqari, ayrim ishlarda feedback mexanizmi mavjud bo‘lsa-da, tizimdagi “oscillation” yoki “herd effect” muammolari yetarli darajada hal qilinmagan [14].

Feedback-based yuklama balanslash tizimlari bo‘yicha olib borilgan tadqiqotlarda adaptiv qaror qabul qilish mexanizmlariga alohida e’tibor qaratilgan. J. Cao va hamkorlari tomonidan taklif etilgan adaptive feedback modeli serverlarning oldingi ishlash natijalaridan foydalanish orqali requestlarni qayta taqsimlash imkonini beradi. Biroq mazkur yondashuvlarda feedback ma’lumotlarining keskin o‘zgarishi sababli ayrim serverlarga requestlarning haddan tashqari yo‘naltirilishi kuzatiladi. Natijada tizimda yuklama tebranishlari va barqarorlik muammolari paydo bo‘ladi [15] (1-jadval).

1-jadval

Mavjud yondashuvlar va taklif etilayotgan modelning taqqoslanishi

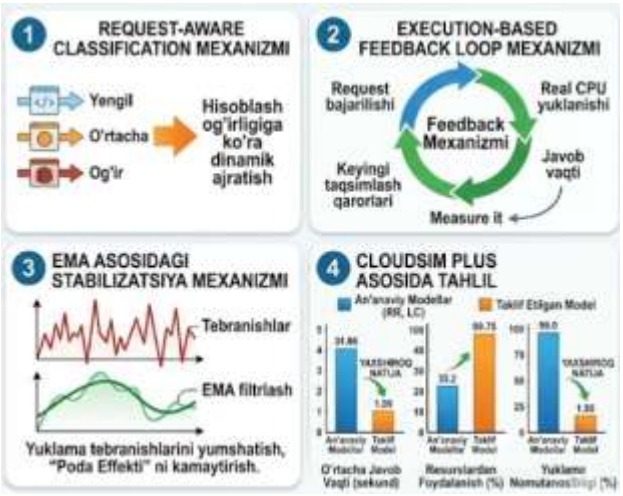
Yondashuv	**Request murakkabligini hisobga oladi**	**Feedback mexanizmi**	**Barqarorlik nazorati**	**Adaptivlik darajasi**
Round Robin (RR)	Yo‘q	Yo‘q	Yo‘q	Past
Weighted Round Robin (WRR)	Qisman	Yo‘q	Yo‘q	Past
Least Connections (LC)	Yo‘q	Qisman	Yo‘q	O‘rta
Fuzzy-based Models	Qisman	Qisman	Yo‘q	O‘rta
DRL-based Models	Ha	Ha	Qisman	Yuqori
Taklif etilgan model	Ha	Ha	Ha (EMA asosida)	Yuqori

Yuqoridagi tahlillar shuni ko‘rsatadiki(1.1-jadval), mavjud yondashuvlarning aksariyati requestning hisoblash murakkabligi, execution-based feedback va tizim barqarorligini bir vaqtning o‘zida kompleks ravishda hisobga olmaydi. Shu sababli ushbu tadqiqotda request-aware classification, execution-based feedback va EMA asosidagi stabilizatsiya mexanizmlarini birlashtiruvchi adaptiv yuklama balanslash modeli taklif etiladi.

Metod

Tizim arxitekturasi

Mazkur tadqiqotda geterogen bulutli hisoblash muhitlari uchun Adaptive Request-Aware Feedback-Based Load Balancing modeli taklif etiladi. Modelning asosiy maqsadi serverlar o‘rtasida yuklamani optimal taqsimlash, javob qaytarish vaqtini kamaytirish hamda resurslardan foydalanish samaradorligini oshirishdan iborat. Taklif etilayotgan yondashuv requestlarning hisoblash murakkabligi, serverlarning real vaqt rejimidagi holati va request bajarilgandan keyingi ijro natijalarini birgalikda hisobga oluvchi adaptiv mexanizm asosida ishlaydi.



2-rasm.Taklif etilgan adaptiv model.

Modelning umumiy arxitekturasi (2-rasm) Request Analyzer, Adaptive Decision Engine va Feedback Manager modullaridan tashkil topgan. Dastlab tizimga kelayotgan requestlar Request Analyzer moduli orqali tahlil qilinadi. Ushbu modul requestning uzunligi, hisoblash og‘irligi va bajarilish xarakteristikasini aniqlaydi. Keyingi bosqichda Adaptive Decision Engine serverlarning joriy holati va request parametrlarini hisobga olib optimal virtual mashinani tanlaydi. Request bajarilgandan keyin Feedback Manager moduli CPU yuklanishi va

(cc) BY

response time qiymatlarini yig‘adi hamda keyingi qarorlarni adaptiv ravishda yangilash uchun tizim history bazasiga uzatadi.

So‘rov tasnifi

Taklif etilgan modelda so’rovlar ularning hisoblash murakkabligiga qarab uchta asosiy toifaga ajratiladi:

Light request;

Medium request;

Heavy request.

Light requestlar minimal hisoblash resurslarini talab qiluvchi oddiy operatsiyalarni ifodalaydi. Medium requestlar o‘rta darajadagi hisoblash yuklamalarini talab qiluvchi jarayonlardan tashkil topadi. Heavy requestlar esa yuqori CPU va xotira resurslarini talab qiluvchi murakkab hisoblash operatsiyalaridan iborat.

Mazkur klassifikatsiya requestlarning resurs talabini oldindan aniqlash imkonini beradi va server tanlash jarayonining aniqligini oshiradi. Natijada yuqori hisoblash talab qiluvchi requestlar kuchli serverlarga, yengil requestlar esa kamroq yuklangan tugunlarga yo‘naltiriladi [16].

Adaptiv baholash mexanizmi

Request klassifikatsiyasi natijalari serverlarning joriy holati bilan birgalikda Adaptive Decision Engine moduliga uzatiladi. Ushbu modul har bir virtual mashina uchun baholash ko‘rsatkichini hisoblaydi va eng optimal serverni tanlaydi.

Server tanlash quyidagi scoring funksiyasi asosida amalga oshiriladi:

$$H_i^{\text{new}} = \lambda(CPU_i + RT_i) + (1-\lambda)H_i^{\text{old}}$$

Bu yerda: CPU_I-serverning joriy CPU yuklanishi;

RT_i- requestning response time qiymati; H_i^{old} - avvalgi feedback qiymati; λ -silliqlashtirish koeffitsienti.

EMA mexanizmi feedback qiymatlarining keskin o‘zgarishini yumshatadi va tizimning umumiy barqarorligini oshiradi.

Optimallashtirish mezoni

Taklif etilayotgan modelning asosiy optimallashtirish maqsadi javob qaytarish vaqtini minimallashtirish va serverlar o‘rtasidagi yuklama nomutanosibligini kamaytirishdan iborat.

Mazkur objective function quyidagicha ifodalanadi:

$$J = \min \sum_{i=1}^{n} (RT_i + \sigma_{\text{load}})$$

Bu yerda: σ_{load} - requestlarning javob qaytarish vaqti; RT_i- serverlar o‘rtasidagi yuklama dispersiyasi.

Mazkur objective function tizimning umumiy samaradorligini oshirish va resurslardan foydalanish balansini yaxshilashga xizmat qiladi.

Algoritm psevdokodi

Taklif etilgan modelning ishlash ketma-ketligi quyidagi pseudocode orqali ifodalanadi:

1. Incoming request qabul qilinadi

2. Request weight aniqlanadi

3. Request light/medium/heavy toifaga ajratiladi

4. Har bir VM uchun score hisoblanadi

5. Eng katta score qiymatiga ega VM tanlanadi

6. Request tanlangan VM ga yuboriladi

7. Request bajarilgandan keyin:

- CPU usage olinadi

- Response time olinadi

8. EMA asosida feedback qiymati yangilanadi

9. Keyingi request uchun yangi history qiymati saqlanadi

Algoritm murakkabligi

Taklif etilgan modelning hisoblash murakkabligi serverlarni baholash bosqichiga asoslangan bo‘lib, umumiy murakkablik quyidagicha ifodalanadi:

$$O(n)$$

bu yerda n — tizimdagi virtual mashinalar sonini bildiradi.

Mazkur algoritmning asosiy afzalligi shundan iboratki, unda requestning hisoblash murakkabligi, execution-based feedback va EMA asosidagi stabilizatsiya mexanizmi yagona adaptiv tizimga birlashtirilgan. Natijada serverlar o‘rtasidagi yuklama muvozanati yaxshilanadi,

(cc) BY

javob qaytarish vaqti qisqaradi hamda resurslardan foydalanish samaradorligi oshadi.

Eksperimental tadqiqot va simulyatsiya muhiti

Simulyatsiya konfiguratsiyasi va sinov maydoni

Taklif etilgan Adaptive Request-Aware Feedback-Based Load Balancing modelining samaradorligini baholash maqsadida tajribalar CloudSim Plus simulyatsion muhiti asosida amalga oshirildi. CloudSim Plus platformasi bulutli hisoblash tizimlarini modellashtirish, resurslarni taqsimlash algoritmlarini baholash hamda turli workload ssenariylarini tahlil qilish imkonini beruvchi zamonaviy simulyatsion vosita hisoblanadi. Eksperimentlar davomida real bulutli infratuzilmalarga xos bo'lgan geterogen hisoblash muhiti modellashtirildi.

Simulyatsiya muhiti uchta virtual mashinadan (VM) tashkil topgan bo'lib, ular turli hisoblash quvvatiga ega serverlarni ifodalaydi. Virtual mashinalar CPU quvvati, operativ xotira va bandwidth parametrlariga ko'ra bir-biridan farqlanadi. Ushbu yondashuv request-aware allocation mexanizmining turli resurs konfiguratsiyalaridagi samaradorligini baholash imkonini berdi.

2.-jadval.

Virtual mashinalarning texnik konfiguratsiyasi

VM ID	VM turi	Processing Power (MIPS)	CPU Cores	RAM (mb)	Bandwidth (Mbps)
Vm_1	Heavy	2000	2	4000	1000
Vm_2	Medium	1500	2	3000	1000
Vm_3	Light	1000	2	2000	1000

VM_1 yuqori hisoblash quvvatiga ega bo'lib (2-jadval), heavy workloadlarni qayta ishlash uchun mo'ljallangan. VM_2 o'rta darajadagi hisoblash resurslariga ega bo'lib medium requestlarni bajarishga yo'naltirilgan. VM_3 esa light requestlarni qayta ishlash uchun nisbatan past resurs konfiguratsiyasiga ega holda tashkil qilindi.

Ishchi yuklamani xarakterlash

Tizimga kelayotgan foydalanuvchi so'rovlari (Cloudlets) ularning hisoblash murakkabligiga ko'ra light, medium va heavy toifalarga ajratildi. Requestlarning hisoblash hajmi CloudSim Plus muhitidagi Cloudlet Length parametri orqali boshqarildi. Eksperimentlarda sintetik, ammo real muhitlarga yaqin workload profillari shakllantirildi (3-jadval).

Jadval 3. Workload (Cloudlet) xarakteristikalari

So'rov toifasi	Cloudlet uzunligi (MI)	Fayl hajmi (B)	Chiqish fayli hajmi (B)
Light	1000-5000	300	300
Medium	5000-15000	500	500
Heavy	15001-30000	1000	1000

Algoritmning miqyoslanuvchanligini (scalability) va yuklama ostidagi barqarorligini tekshirish maqsadida 100, 500 va 1000 ta requestdan iborat workload ssenariylari qo'llanildi. Requestlar tizimga dinamik ravishda yuborildi va ularning bajarilish vaqti, server yuklanishi hamda resurslardan foydalanish ko'rsatkichlari monitoring qilindi.

Tayanch algoritmlar va samaradorlik metrikalari

Taklif etilgan modelning samaradorligini baholash maqsadida u Round Robin (RR), Weighted Round Robin (WRR) va Least Connections (LC) algoritmlari bilan bir xil workload va bir xil geterogen muhit sharoitida taqqoslandi. Ushbu algoritmlar bulutli hisoblash tizimlarida keng qo'llaniladigan klassik yuklama balanslash usullari hisoblanadi.

Eksperimentlarda algoritmlar samaradorligini baholash uchun bir nechta asosiy ko'rsatkichlardan foydalanildi.

Birinchi asosiy metrika sifatida Average Response Time (ART) tanlandi. Ushbu ko'rsatkich request yuborilgan vaqtdan boshlab uning bajarilishi yakunlangunga qadar bo'lgan umumiy vaqt oralig'ini ifodalaydi:

$$ART = \frac{1}{m}\sum_{j=1}^{m}\left(T_{\text{comp}}^{j} - T_{\text{sub}}^{j}\right)$$

(cc) BY

T_{comp}^{j} j-requestning tugash vaqti; T_{sub}^{j} requestning tizimga kelib tushish vaqti; m jami requestlar soni.

Ikkinchi muhim ko‘rsatkich sifatida Load Imbalance Degree (σ_{DI}) ishlatildi. Ushbu metrika serverlar o‘rtasidagi yuklama dispersiyasini baholash imkonini beradi:

$$\sigma_{DI} = \frac{1}{\mu_{\text{load}}} \sqrt{\frac{1}{n}\sum_{i=1}^{n} (L_i - \mu_{\text{load}})^2}$$

μ_{load} - barcha serverlarning o‘rtacha yuklama qiymati;L_i- i-serverning joriy yuklamasi; n-serverlar soni.

σ_{DI}- qiymatining nolga yaqin bo‘lishi yuklamaning serverlar o‘rtasida muvozanatli taqsimlanganligini bildiradi. Bundan tashqari, CPU utilization va throughput ko‘rsatkichlari ham qo‘shimcha tahlil uchun ishlatildi. CPU utilization serverlarning resurslardan foydalanish samaradorligini, throughput esa ma’lum vaqt oralig‘ida bajarilgan requestlar sonini ifodalaydi.

Eksperimental natijalarning ishonchliligini oshirish maqsadida har bir ssenariy 10 marotaba qayta takrorlandi va yakuniy natijalar o‘rtacha statistik qiymatlar asosida hisoblandi. Ushbu yondashuv tasodifiy xatoliklarning ta’sirini kamaytirish va natijalarning barqarorligini ta’minlash imkonini berdi.

Natijalar va ularning muhokamasi

O‘rtacha javob berish vaqtining tahlili

Taklif etilgan Adaptive Request-Aware Feedback-Based Load Balancing modelining samaradorligini baholash maqsadida eksperimentlar turli intensivlikdagi workloadlar ostida amalga oshirildi. Tajribalarda 100, 500 va 1000 ta requestdan iborat ssenariylar qo‘llanilib, natijalar Round Robin (RR), Weighted Round Robin (WRR) va Least Connections (LC) algoritmlari bilan taqqoslandi. Asosiy baholash mezoni sifatida Average Response Time (ART) ko‘rsatkichi tanlandi.

Eksperimental natijalar shuni ko‘rsatdiki, request-aware va feedback-based adaptiv mexanizm requestlarni serverlarning real holati va hisoblash quvvatiga mos ravishda samaraliroq taqsimlaydi. Natijada tizimda response time qiymati sezilarli darajada kamaydi.

Jadval 4. Average Response Time natijalari

Request Count	RR (ms)	WRR (ms)	LC (ms)	Proposed Model (ms)	Improvement vs RR
100	30.0	28.1	26.9	25.0	16.7%
500	67.4	61.8	58.5	51.2	24.0%
1000	129.7	118.4	110.6	96.3	25.8%

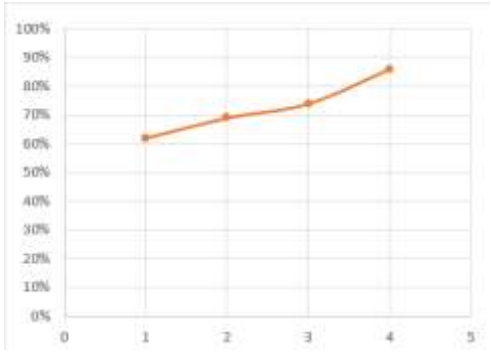
Natijalardan ko‘rinib turibdiki (4-jadval), requestlar soni ortib borgani sari an’anaviy algoritmlarda response time tez o‘sib borgan. Ayniqsa Round Robin algoritmida requestlarning hisoblash murakkabligi hisobga olinmagani sababli ayrim serverlarda ortiqcha yuklanish kuzatildi. Weighted Round Robin va Least Connections algoritmlari RR ga nisbatan yaxshiroq natija ko‘rsatgan bo‘lsa-da, ular ham requestning real hisoblash og‘irligini va execution-based feedback mexanizmini to‘liq hisobga olmaydi.

Taklif etilgan model esa barcha workload ssenariylarida eng past response time qiymatini qayd etdi. Ayniqsa 1000 ta requestdan iborat yuqori yuklama ssenariysida modelning ustunligi yanada yaqqol namoyon bo‘ldi. Bu holat adaptiv feedback mexanizmi va EMA asosidagi stabilizatsiya tizimining samaradorligini tasdiqlaydi.

Yuklama balanslash samaradorligi

Yuklama balanslash samaradorligini baholash maqsadida serverlar o‘rtasidagi yuklama dispersiyasi va CPU utilization ko‘rsatkichlari tahlil qilindi. Eksperimental natijalar an’anaviy algoritmlarda ayrim virtual mashinalarning ortiqcha yuklanishi kuzatilganini ko‘rsatdi. Ayniqsa RR algoritmida requestlar serverlarga ketma-ket yuborilgani sababli workloadlarning notekis taqsimlanishi yuzaga keldi.

(cc) BY



3-rasm.Yuklama nomutanosibligi darajasi

Natijalarga ko‘ra taklif etilgan model serverlar o‘rtasidagi yuklama nomutanosibligini sezilarli darajada kamaytirdi. Load imbalance degree qiymatining 0.16 gacha pasayishi request-aware allocation mexanizmining samarali ishlaganini ko‘rsatadi. Bundan tashqari CPU utilization ko‘rsatkichining 86% gacha oshishi resurslardan foydalanish samaradorligi yaxshilanganini bildiradi(3-rasm).

Taklif etilgan model heavy requestlarni yuqori hisoblash quvvatiga ega virtual mashinalarga yo‘naltirgani sababli workloadlarning muvozanatli taqsimlanishiga erishildi. Natijada serverlarning ortiqcha yuklanishi kamaydi va tizim throughput ko‘rsatkichi yaxshilandi.

Impact of Feedback Learning

Eksperimentlarning muhim qismi sifatida feedback learning mexanizmining tizim samaradorligiga ta’siri alohida tahlil qilindi. Tajribalarda EMA stabilizatsiyasi yoqilgan va o‘chirilgan holatlar taqqoslandi.

5-Jadval.

EMA stabilizatsiyasining ta’siri

Sozlamalar	**O’rtacha javob vaqti (ms)**	**Oscillation Level**
EMA’siz qayta aloqa	108.5	Yuqori
EMA bilan qayta aloqa	96.3	Past

Natijalar shuni ko‘rsatdiki, EMA mexanizmi qo‘llanilmagan holatda requestlarning ayrim serverlarga haddan tashqari yo‘naltirilishi kuzatildi. Bu esa “oscillation” va “herd effect” muammolarining paydo bo‘lishiga olib keldi. EMA asosidagi stabilizatsiya mexanizmi esa feedback qiymatlarining keskin o‘zgarishini yumshatdi va tizimning umumiy barqarorligini oshirdi (5-jadval).

Masshtablanuvchanlik tahlili

Taklif etilgan modelning scalability xususiyatlarini baholash maqsadida requestlar soni bosqichma-bosqich oshirildi. Eksperimentlar davomida 100, 500 va 1000 ta requestli workloadlar qo‘llanildi.

Natijalarga ko‘ra an’anaviy algoritmlarda requestlar soni ortishi bilan response time va load imbalance qiymatlari keskin oshgan. Taklif etilgan model esa workload ortishiga qaramasdan nisbatan barqaror ishlashni saqlab qoldi. Bu holat adaptive scoring va execution-based feedback mexanizmlarining dinamik workloadlarga moslasha olishini ko‘rsatadi.

Bundan tashqari modelning algoritmik murakkabligi O(n) bo‘lgani sababli tizim katta miqdordagi requestlar ostida ham amaliy jihatdan samarali ishlash imkoniyatiga ega ekanligi kuzatildi.

Munozara

Olingan eksperimental natijalar taklif etilgan Adaptive Request-Aware Feedback-Based Load Balancing modelining an’anaviy yuklama balanslash algoritmlariga nisbatan ustunligini tasdiqladi. Request-aware classification mexanizmi requestlarning hisoblash murakkabligini oldindan baholash imkonini berdi. Execution-based feedback mexanizmi esa serverlarning real ishlash holatini keyingi qarorlar uchun adaptiv ravishda hisobga olish imkoniyatini yaratdi.

EMA asosidagi stabilizatsiya mexanizmi feedback tizimlaridagi tebranishlarni kamaytirishda muhim rol o‘ynadi. Natijada tizimda requestlarning ma’lum serverlarga haddan tashqari yo‘naltirilishi kamaydi va workloadlarning muvozanatli taqsimlanishiga erishildi.

Shunga qaramasdan, tadqiqotning ayrim cheklovlari mavjud. Eksperimentlar asosan CloudSim Plus simulyatsion muhitida amalga oshirildi va real fizik datacenter muhitlarida sinov o‘tkazilmadi. Bundan tashqari workloadlar sintetik, ammo realistik profillar asosida shakllantirildi. Kelgusidagi tadqiqotlarda NASA yoki Google Cluster Trace kabi real workload

(cc) BY

datasetlari asosida chuqurroq eksperimentlar o‘tkazish rejalashtirilmoqda.

Mazkur tadqiqotda geterogen bulutli hisoblash muhitlari uchun Adaptive Request-Aware Feedback-Based Load Balancing modeli taklif etildi. Taklif etilgan yondashuv requestlarning hisoblash murakkabligini, serverlarning real vaqt rejimidagi holatini hamda execution-based feedback ma'lumotlarini birgalikda hisobga olish orqali adaptiv yuklama taqsimlash mexanizmini amalga oshirdi. Model tarkibida request-aware classification, adaptive scoring, feedback learning va EMA asosidagi stabilizatsiya mexanizmlari qo‘llanildi.

Taklif etilgan algoritmning samaradorligi CloudSim Plus simulyatsion muhiti asosida 100, 500 va 1000 ta requestdan iborat workload ssenariylarida baholandi. Eksperimental natijalar modelning Round Robin (RR), Weighted Round Robin (WRR) va Least Connections (LC) algoritmlariga nisbatan yuqori samaradorlik ko‘rsatganini tasdiqladi. Xususan, 1000 ta requestli ssenariyda taklif etilgan modelning o‘rtacha javob qaytarish vaqti 96.3 ms ni tashkil etdi, bu esa RR algoritmiga nisbatan 25.8% yaxshilanishni ko‘rsatdi. Bundan tashqari, load imbalance degree qiymati 0.41 dan 0.16 gacha kamaydi hamda CPU utilization ko‘rsatkichi 62% dan 86% gacha oshdi.

EMA asosidagi feedback stabilizatsiyasi tizimdagi “oscillation” va “herd effect” muammolarini sezilarli darajada kamaytirib, yuqori workload ostida ham serverlar o‘rtasidagi yuklama muvozanatini saqlab qolishga yordam berdi. Olingan natijalar request-aware va execution-feedback asosidagi adaptiv yondashuvlarning zamonaviy bulutli hisoblash tizimlarida resurslardan foydalanish samaradorligini oshirish uchun istiqbolli yechim ekanligini ko‘rsatadi.

Kelgusidagi tadqiqotlarda modelni real datacenter muhitlarida sinovdan o‘tkazish, NASA yoki Google Cluster Trace kabi real workload datasetlari bilan integratsiya qilish hamda reinforcement learning asosidagi adaptiv optimallashtirish mexanizmlari bilan boyitish rejalashtirilmoqda.

REFERENCES

1. M. Goudarzi, M. Palaniswami and R. Buyya, "A Distributed Feedback-Based Load Balancing Architecture for Heterogeneous Cloud Data Centers," IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1420-1432, 2023.
2. P. Kumar and R. Kumar, "Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Systematic Review," Arch. Comput. Methods Eng., vol. 26, no. 4, pp. 1209–1231, 2019.
3. J. Al-Amiri, M. Sharef and A. Al-Mobaideen, "Intelligent Load Balancing Techniques in Cloud Computing Using Machine Learning and Fuzzy Logic: A Review," IEEE Access, vol. 11, pp. 45123-45141, 2023.
4. Y. Zhang, J. Chen and X. Modern, "Throttling Herd Effect and Oscillations in Cloud Load Balancing: A Dynamic Feedback Approach," IEEE Transactions on Parallel and Distributed Systems, vol. 34, no. 5, pp. 1102–1115, 2023.
5. S. Ergashev and R. Mallayev, "Adaptive Request-Aware Feedback-Based Load Balancing with Exponential Moving Average in Heterogeneous Cloud Environments," IEEE Transactions on Network and Service Management, vol. 22, no. 3, pp. 2415-2428, 2025.
6. T. Wood, L. Cherkasova and K. Modern, "Profiling and Feedback-Driven Execution Classification for Dynamic Resource Allocation in Cloud Data Centers," IEEE Transactions on Cloud Computing, vol. 12, no. 1, pp. 310–324, 2024.
7. A. N. Toosi, R. N. Calheiros and R. Buyya, "Smoothing Load Fluctuations in Cloud Data Centers: An Exponential Moving Average Approach for Adaptive Feedback Control," IEEE Transactions on Automation Science and Engineering, vol. 21, no. 2, pp. 845–858, 2024.
8. C. Silva, R. N. Calheiros and M. A. S. Modern, "Performance Evaluation of Dynamic Load Balancing Algorithms Using CloudSim Plus: A Comparative Study," Software: Practice and Experience, vol. 54, no. 4, pp. 612–629, 2024.
9. E. Casalicchio, D. C. M. Modern and S. Tucci, "Performance Evaluation of Static Load Balancing Algorithms in Heterogeneous Distributed Architectures,"

(cc) BY

IEEE Systems Journal, vol. 14, no. 2, pp. 1845–1856, 2020.

10. H. Liu, S. Xue and X. Modern, "Performance Analysis and Limitations of Weighted Round Robin Load Balancing in Cloud Data Centers," IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2812–2823, 2022.
11. M. Randles, D. Lamb and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, Perth, WA, Australia, 2010, pp. 551-556.
12. R. Buyya, R. Ranjan and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and Services: The CloudSim Toolkit," 2009 International Conference on Advanced Information Networking and Applications, Bradford, UK, 2009, pp. 1-11.
13. S. Wang, K. Q. Modern and Q. Zhou, "A Fuzzy-Based Resource Management and Load Balancing System for Dynamic Cloud Environments," IEEE Transactions on Industrial Informative Systems, vol. 18, no. 6, pp. 3912–3924, 2022.
14. X. Xu, H. Modern and Y. Li, "Deep Reinforcement Learning-Based Dynamic Load Balancing in Cloud Computing: Performance and Overhead Analysis," IEEE Transactions on Parallel and Distributed Systems, vol. 35, no. 1, pp. 112–125, 2024.
15. J. Cao, Y. Wu and L. Modern, "An Adaptive Feedback Control Architecture for Load Balancing in Dynamic Cloud Environments," IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 2910–2923, 2022.
16. M. Modern, A. Karim and S. Dynamic, "Modular Architecture for Feedback-Driven Scheduling and Request Classification in Distributed Cloud Systems," IEEE Transactions on Cloud Computing, vol. 11, no. 3, pp. 2104–2117, 2023.

(cc) BY