

UOT 004.451.44

Əliquliyev R.M.¹, İmamverdiyev Y.N.², Abdullayeva F.C.³

AMEA İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan

¹rasim@science.az, ²yadigar@iit.science.az, ³farqana@iit.ab.az

BULUD TEXNOLOGİYALARINDA YÜK BALANSLAŞDIRILMASI ÜÇÜN ÇOXKRİTERİYALI OPTİMALLAŞDIRMA ÜSULU

Bulud texnologiyaları mühitində tapşırıqların planlaşdırılması prosesinin optimallaşdırılması çoxkriteriyalı NP-çətin (ing. NP-hard, nondeterministic polynomial) problemdir. Məqalədə bulud mühitində yükün balanslaşdırılması üçün “sürü” intellektinə əsaslanan optimallaşdırma alqoritminə (ing. Particle Swarm Optimization, PSO) əsaslanan çəkili α PSO – TBLB (ing. Task Based Load Balancing) üsulu təklif edilmişdir. Üsul bulud texnologiyaları mühitində virtual maşınlarda izafi yükləmə yaradan tapşırıqların yüklənmiş virtual maşınlardan münasib virtual maşınlara optimal miqrasiyasını təmin edir. Təklif edilmiş optimallaşdırma üsulunda tapşırıqların emal vaxtı və ötürülmə vaxtının minimallaşdırılması məqsəd funksiyaları kimi seçilmişdir. Təklif edilmiş yanaşmanın eksperimental yoxlanması Cloudsim və Jswarm proqramlarında aparılmışdır. Simulyasiya nəticəsində təklif edilmiş üsul əsasında tapşırıqların planlaşdırılması üçün optimal həll tapılmış, tapşırıqların Virtual Maşınlara (VM-lərə) bərabər paylanması təmin edilmiş və onların VM-lərə təyin edilməsi prosesinə olduqca az vaxtın sərf edilməsi əldə olunmuşdur.

Açar sözlər: *Bulud texnologiyaları, “sürü” intellektinə əsaslanan optimallaşdırma alqoritm, virtual maşın miqrasiyası, tapşırıqların planlaşdırılması, Cloudsim, Jswarm, verilənlər intensivliyi, hesablama intensivliyi.*

Giriş

Bulud verilənlər mərkəzi şəbəkələşmiş serverlər toplusudur, böyük çeşiddə hesablama resursları ilə təchiz edilmiş heterogen hostlardan ibarətdir [1]. Bulud verilənlər mərkəzində virtuallaşma texnologiyası server heterogenliyini aradan qaldırır, server konsolidasiyasını təmin edir, serverdən istifadənin effektivliyini artırır [2]. Virtuallaşmış host müxtəlif texniki tələbli resursları və zamana görə dəyişən müxtəlif tipli iş yükü olan çoxsaylı VM-lərin hostinqini təmin edə bilər. İş yükü gözlənilmədən dəyişkən olan heterogen VM-lərin hostinqini təmin edən serverlər host daxilində resurs disbalansının yaranmasına səbəb ola bilər. Bu vəziyyət məhsuldarlığın azalmasına, nəticədə isə servis səviyyəsi müqavilənin pozuntusuna gətirib çıxarır.

Bulud texnologiyalarında yük balanslaşdırılmasını təmin etmək üçün VM-lərin serverlər arasında miqrasiyasını həyata keçirirlər. VM miqrasiyasını işlək vəziyyətdə olan VM-ləri bir fiziki maşından digər maşına köçürməklə təmin edirlər. Bu zaman hesablama gücünün artırılması, yaddaşın artırılması, sürətli kommunikasiya, enerjiyə qənaət kimi göstəricilər əldə edirlər. Bulud texnologiyaları mühitində yüklənmiş VM-lərin miqrasiyası əsasında sistemin yük balanslaşdırılmasının təmin edilməsi ənənəvi yanaşmadır. Bu yanaşma yaddaş yüklənməsi (ing. dirty memory), həm fiziki maşında, həm də host fiziki maşınında böyük yaddaşın tutulması, VM-in işində fasilənin yaranması, miqrasiya prosesinə çox vaxtın və xərclərin sərf olunması kimi problemlər meydana gətirir. Ənənəvi yanaşmaların bu mövcud problemlərini aradan qaldırmaq məqsədi ilə VM miqrasiyası əvəzinə VM-də yük artıqlığı yaradan tapşırıqların az yüklənmiş VM-lərə miqrasiyasını həyata keçirirlər [3]. Bulud texnologiyaları mühitində tapşırıqlar iki qrupa bölünür [4]: hesablama intensivliyi (ing. computing intensive) tələb edən tapşırıqlar; verilənlər intensivliyi (ing. data intensive) tələb edən tapşırıqlar. Verilənlər intensivliyi tələb edən tapşırıqların planlaşdırılması məsələsində adətən verilənlərin miqrasiyasının sayını azaltmağa çalışırlar, beləliklə, verilənlərin ötürülmə vaxtının azaldılmasına nail olurlar. Hesablama intensivliyi tələb edən tapşırıqların planlaşdırılması məsələsində isə planlaşdırma strategiyası verilənləri məhsuldarlığı yüksək olan kompüterlərə miqrasiya etdirir, bu yolla tapşırıqın emal vaxtını azaltmış olurlar. Bulud mühitində VM – özünün əməliyyat sistemi, yaddaşı, proqram

təminatı olan izolyasiya edilmiş iş mühitidir. Bulud mühitində yük balanslaşdırılmasının təmin edilməsi istiqamətində çox sayda tədqiqatlar aparıldığına baxmayaraq, burada əlavə tapşırıqların yüklənmiş VM-dən digər VM-lərə paylanması üçün optimal həllin tapılması məsələsi problem olaraq qalır. Tapşırıqların planlaşdırılması NP-çətin problem sayıldığından bu məsələni həll etmək üçün evristik alqoritmlərdən istifadə edirlər [5]. Təqdim olunan məqalədə tapşırıqların planlaşdırılması PSO alqoritmindən istifadə etməklə təmin olunur.

Təklif edilən üsulun əsas ideyası hesablama intensivliyi tələb edən tapşırıqları məhsuldarlığı yüksək olan VM-ə miqrasiya etdirməkdir. Bundan əlavə, mövcud alqoritmlərdə planlaşdırma prosesində minimallaşdırılacaq kriteriyaların çəkisi eyni götürüldüyündən, məqsəd funksiyasının daha optimal olmasını tənzimləmək mümkün olmur. Bu səbəbdən planlaşdırma prosesində kriteriyaların vaciblik dərəcəsini göstərmək üçün onlara müvafiq çəki əmsalları verilməlidir. Burada çəki əmsalları planlaşdırma prosesində kriteriyaların vacibliyini göstərmək üçün istifadə olunur. Kriteriyalara çəkilərin mənimsədilməsi daha yaxşı optimal həll tapmağa imkan verir. Məqalənin məqsədi kriteriyalara çəkilər mənimsədildikdə daha yaxşı optimal həll tapıldığını nümayiş etdirməkdir. Məqalədə təqdim olunan digər mühüm yanaşma tapşırıqların planlaşdırılmasının iş axını (*ing. workflow scheduling*) şəklində həyata keçirilməsidir. Bu məqsədlə iş axını istiqamətlənmiş atsiklik periodik olmayan qraf şəklində (*ing. Directed Acyclic Graph*) təsvir edilir. Burada qrafın təpələrini tapşırıqlar, uyğun təpələr arasındakı tilləri isə tapşırıqlar arasındakı asılılıq təşkil edir. [3]-də PSO alqoritmini tətbiq etməklə əlavə tapşırıqların yeni host VM-lərə miqrasiyasını təmin edən üsul təklif edilir. Burada bizim təklif etdiyimiz üsula yaxın yanaşma irəli sürülür. Lakin aparılan eksperimentlərdən aydın şəkildə görünür ki, burada yükün paylanmasında ciddi disbalans vardır. Bundan əlavə üsulda tapşırıqların planlaşdırılmasının optimallaşdırılması da çox uğurla aparılmamışdır, optimal həllin tapılmasına çox vaxt sərf olunur.

Məqalədə tapşırıqların planlaşdırılmasının yuxarıdakı problemlərini (planlaşdırmaya sərf olunan vaxtın minimallaşdırılması, yükün VM-lər arasında bərabər paylanması, asılı tapşırıqların miqrasiyası) aradan qaldırmaq məqsədi ilə çəkili PSO alqoritmi əsasında qurulmuş α PSO – TBLB adlı çoxkriteriyalı optimallaşdırma üsulu təklif edilir. Üsulda tapşırıqların emal vaxtının və ötürülmə vaxtının minimallaşdırılması optimallaşdırma kriteriyaları kimi seçilmişdir. İşin əsas yeniliklərinə aşağıdakılar aiddir:

- Bulud mühitində tapşırıqların çəkili planlaşdırılması üsulu təklif olunur. Burada optimallaşdırma məsələsi iki kriteriyanın minimallaşdırılması hesabına təmin olunur. Üsul tapşırıqların emal vaxtının və ötürülmə vaxtının minimallaşdırılmasını əsas hədəfə alır.

- Təklif edilmiş üsula əsasən tapşırıqların planlaşdırılması üçün PSO alqoritmi qurulur.

- Təklif edilmiş üsulun imkanları Jswarm və Cloudsim proqram paketlərində qiymətləndirilir.

Buludda yük balanslaşdırılması üçün planlaşdırma üsullarının tipləri və məqsədləri

Planlaşdırma nəzəriyyəsi XX əsrin 50-ci illərində tədqiq olunmağa başlanmışdır. Planlaşdırma nəzəriyyəsi – tapşırıqlar çoxluğunun yerinə yetirilməsinin ardıcılığını qurmaq (nizamlamaq) lazım gələn məsələləri tədqiq edir. Nizamlama məsələsi ümumi xarakter daşıyır. Bu tip məsələlər tapşırıqların yerinə yetirilməsində bu və ya digər növün seçilməsinə zərurət yarandığı sahələrdə (məsələn, istehsalatda tapşırıqların paylanması, təyyarələrin eniş planının qurulması, qatarların hərəkət planının qurulması, xidmət sistemlərində müştərilərə xidmət olunması və s.) meydana çıxır.

Plan (ing. shedule) – hər bir i maşınına və t vaxtına müəyyən bir tapşırıq qarşı qoyan funksiyaya deyilir. Bu funksiyanın müxtəlif təsvir formaları vardır [6].

Planların təsvir formaları da fərqlidir. Planlar aşağıdakı üsullarla təsvir oluna bilər: düsturlar, cədvəllər, qant diaqramı (*ing. gantt chart*). Optimal planların qurulması məsələsini təsvir etmək üçün üç hissədən ibarət $\alpha/\beta/\gamma$ yazılışı istifadə olunur. Burada α – xidmət sistemi, β – tapşırıqların xarakteristikaları, γ – optimallaşdırma kriteriyaları. Bulud mühitində tapşırıqların planlaşdırılmasına və yük balanslaşdırılmasına aşağıdakı kimi tərif verirlər [7, 8]:

Tapşırıqların planlaşdırılması – tapşırıqların VM-lərə təyin edilməsinin optimallaşdırılması.

Yük balanslaşdırılması (ing. *load balancing*) – hesablama yükünün resurslar arasında paylanması həyata keçirən istənilən əməliyyat.

Yük balanslaşdırılması – müştərilərin sorğularını bir neçə server arasında paylayan sistem.

Planlaşdırma üsulları aşağıdakı kateqoriyalara bölünür [9, 10]:

- *İstifadəçi səviyyəsində* (ing. *user level*). Qurulmuş üsul bulud provayderi ilə istifadəçi arasında servisin təqdim edilməsi zamanı meydana çıxan problemi aradan qaldırmağa xidmət edir.

- *Sistem səviyyəsində* (ing. *system level*). Bulud verilənlər mərkəzində resursların idarə edilməsinə xidmət edir.

- *Statik planlaşdırma* (ing. *static scheduling*). Tapşırıqların emalı əvvəlcədən müəyyən olunur. Burada hər bir tapşırığın yerinə yetirilmə vaxtı və ardıcılığı əvvəlcədən verilir.

- *Dinamik planlaşdırma* (ing. *dynamic scheduling*). Tapşırığın hansı vaxt daxil olacağı haqqında informasiya məlum olmur.

- *Mərkəzləşdirilmiş planlaşdırma* (ing. *centralized scheduling*). Tapşırıqlar toplusunu özündə cəmləşdirən mərkəzi prosessor qurğusundan istifadə olunur. Bu qurğu tapşırıqların digər emal qurğularına ötürülməsini təmin edir.

- *Paylanmış planlaşdırma* (ing. *distributed scheduling*). Sorğuları və tapşırıqları idarə etmək üçün lokal planlaşdırma qurğusundan istifadə edir. Bu üsulun mərkəzləşdirilmiş planlaşdırma üsulu ilə müqayisədə effektivliyi aşağıdır.

- *Fasiləli planlaşdırma* (ing. *preemptive scheduling*). Tapşırığın emal prosesinin dayandırılması və digər resursu miqrasiya etdirilməsinin mümkünlüyü.

- *Fasiləsiz* (ing. *non-preemptive scheduling*) *planlaşdırma*. Tapşırıq yalnız tamamlandıqdan sonra, onu emal edən resurs digər emal prosesi ilə məşğul ola bilər.

- *Onlayn planlaşdırma* (ing. *online scheduling*). Hər bir tapşırıq yalnız bir dəfə planlaşdırılır və planlaşdırmanın nəticəsi dəyişdirilə bilməz.

- *Oflayn planlaşdırma* (ing. *offline scheduling*). Tapşırıqlar daxil olan anda resurslara təyin edilmir, onlar toplanır və təyin olunmaq məqsədi ilə qiymətləndirilir.

- *Tapşırıq səviyyəsində planlaşdırma* (ing. *task-level scheduling*). Lokal bulud verilənlər mərkəzlərində tapşırığın VM-ə təyin edilməsi prosesini optimallaşdırmağa xidmət edir. Məqsədi iş axınının bütün istismar xərclərini QoS göstəricilərinə nəzərən minimallaşdırmaqdır.

- *Servis səviyyəsində planlaşdırma* (ing. *service-level scheduling*). Tapşırığın xidmətə təyin edilməsini həyata keçirir. Burada iş axınının tapşırıqları onların funksional və qeyri-funksional QoS tələblərinə əsasən təyin etdirilir.

Bulud texnologiyalarında yük balansını yanaşmalarında əsasən aşağıdakı kriteriyaların təmin olunmasına çalışırlar [10–13]:

- *Büdcə* (ing. *budget*). Müştərilərin istifadə etdikləri bulud resurslarına ödədikləri xərclərdir.

- *Vaxt limiti* (ing. *deadline*). İş axınının emal olunmasına qoyulmuş vaxt limitidir.

- *Etibarlılıq* (ing. *reliability*). Tapşırığın emalının uğurla başa çatacağını göstərən ehtimaldır. Bu tələbi ödəmək üçün, adətən planlaşdırma metodlarında replikasiya və ehtiyat nüsxə üsullarından istifadə edirlər.

- *Əlçatarlılıq* (ing. *availability*). Bulud resurslarının əlçatarlılığının təmin edilməsi hesabına əldə olunur.

- *Tapşırığın maksimal emal vaxtının minimallaşdırılması* (ing. *minimizing the makespan*). İş axınının sonuncu tapşırığının emalının tamamlanmasına ayrılmış vaxtdır.

- *Servis səviyyəsi müqaviləsinə əsaslanan planlaşdırma* (ing. *Service Level Agreement, SLA*). Müştəri ilə provayder arasında qəbul olunmuş QoS tələblərinin ödənilməsinə xidmət edir.

- *Təhlükəsizlik* (ing. *security*). Təhlükəsiz planlaşdırmanı təmin edir, buluda xas təhlükəsizlik hücumlarının qarşısını almaq imkanına malikdir.

- *Yük balanslaşdırılması (ing. load balancing)*. İstənilən bulud resursundakı yük artıqlığını aradan qaldırmaqla, resursun istifadə olunmasında optimallığı təmin edir.
- *Cavabvermə vaxtı (ing. response time)*. Sistemin qəbul etdiyi sorğunun emalına sərf etdiyi ümumi vaxtı ölçür. Bu vaxtın, əsasən, minimum olmasına çalışırlar.
- *Miqyaslanma (ing. scalability)*. Məhdud sayda prosessorları və maşınları olan sistemin yük balanslaşdırılması alqoritmlərini icra edə bilməsi imkanını müəyyən edir.
- *Resurs utilizasiyası (ing. resource utilization)*. Sistemin resurslarından istifadə dərəcəsini müəyyən edir. Ən yaxşı yük balanslaşdırılması alqoritm maksimum resurs istifadəsinə yol verməlidir.
- *Miqrasiya vaxtı (ing. migration time)*. Bulud sistemində tapşırığın bir maşından digər maşına miqrasiyasına sərf olunan vaxt. Bulud sisteminin məhsuldarlığını artırmaq üçün bu göstəricinin minimallaşdırılmasına nail olmaq lazımdır.
- *Məhsuldarlıq (ing. performance)*. Yük balanslaşdırıldıqdan sonra sistemin keçdiyi vəziyyətin effektivliyini göstərir. Burada hesab edilir ki, sistemin məhsuldarlığı yuxarıdakı parametrlər tam ödəndiyi halda yüksələ bilər.
- *Ötürücülük imkanı (ing. throughput)*. Yerinə yetirilmiş tapşırıqların ümumi sayı. Bu parametrlər yüksək olduqda, sistem bir o qədər məhsuldarlıqlı hesab olunur.
- *Əlaqədar xərclər (ing. associated overhead)*. Yük balanslaşdırılması alqoritminin yerinə yetirilməsinə sərf olunan ümumi xərclər. Xərcin qiymətinin minimum olması alqoritm daha uğurlu olduğunun göstəricisidir.
- *İmtinalara dayanıqlılıq (ing. fault tolerant)*. Alqoritm sistemin istənilən qovşağının sıradan çıxdığı vəziyyətdə də dəqiq işləmə qabiliyyətidir.

Əlaqəli işlər

Tapşırıqların planlaşdırılması və yük balanslaşdırılması üçün qurulmuş alqoritmlər bulud texnologiyaları mühitində SLA-nın pozulması hallarının sayını kifayət qədər azaldır. Yük balanslaşdırılması alqoritmləri dinamik və statik kateqoriyalara bölünürlər. Statik alqoritmlər sistemin cari vəziyyətindən asılı olmayan alqoritmlərdir və burada zəruri şərt sistemin vəziyyəti haqqında əvvəlcədən bilginin olmasıdır. Dinamik alqoritmlər sistemin cari vəziyyətinə bağlı alqoritmlərdir. Statik alqoritmlər qovşağın yükündə az dəyişiklik olduğu halda doğru işləyə bilər. Bulud mühitində yük dəyişməsi müxtəlif zamanlarda baş verdiyi üçün statik alqoritmlər bu mühit üçün yararlı hesab edilmir. Bu səbəbdən bulud kimi böyük hesablama mühitlərində yük balanslaşdırılması üçün statik alqoritmlərin işlənilməsinə elmi cəmiyyət tərəfindən diqqət ayrılır. Bulud texnologiyaları yük balanslaşdırılması mexanizminin optimal olmasına böyük ehtiyac duyur. Bu səbəbdən təqdim olunan tədqiqat işinin bu bölməsində dinamik yük balanslaşdırılması alqoritmlərinin icmalı aparılır.

Dinamik yük balanslaşdırılması alqoritmləri. Bulud texnologiyaları mühitində yük balanslaşdırılmasını təmin edən üsullardan biri tapşırıqların VM-lərə miqrasiya etdirilməsidir. Bunu yük artıqlığı yaradan tapşırıqları az yüklənmiş VM-lərə ötürməklə təmin edirlər [3]. [14]–də tapşırığın emal vaxtı, ötürülmə vaxtı və emal xərclərini minimallaşdırmaqla tapşırıqların planlaşdırılması üçün çoxməqsədli kompleks optimallaşdırma üsulu təklif edilir. Qurulmuş optimallaşdırma üsulunda bir-biri ilə ziddiyyət təşkil edən məqsəd funksiyaları istifadə edilir və optimal həllin tapılması PSO alqoritmünün əsasında aparılır. [15]–də PSO alqoritmünü tətbiq etməklə yük artıqlığı yaradan tapşırıqları yeni VM-lərə miqrasiya etdirən optimallaşdırma üsulu təklif olunur. Üsulda tapşırıqların emal vaxtının və ötürülmə vaxtının minimallaşdırılması əsas optimallaşdırma kriteriyaları kimi götürülür. Burada tapşırıqların miqrasiyası ərzində VM-lər fasilə etmədiyi üçün təklif edilən üsul, həmçinin yaddaşın boş dayanması, yaddaşdan istifadə və xərclərin azaldılması üçün böyük imkanlar yaradır. Üsulun çatışmazlığı onun yüklənmiş VM-ə daxil olan, yalnız bir-birindən asılı olmayan tapşırıqları digər homogen VM-ə miqrasiya etdirməsidir. [16]–də tapşırıqların ötürülmə vaxtı və emal xərcləri, enerji sərfiyyatı, tapşırıqların növbəsinin uzunluğu əsas minimallaşdırma kriteriyaları götürülərək, tapşırıqların planlaşdırılması üçün genetik alqoritm əsasında çoxmeyarlı optimallaşdırma üsulu təklif edilmişdir. Üsulda

sadalanan kriteriyalar əsasında qurulmuş dörd məqsəd funksiyası bir-biri ilə ziddiyyət təşkil edir. Burada emal və enerji xərclərinin minimallaşdırılması əsas hədəf götürüldüyü üçün üsul həm müştəri, həm də provayder tərəfdə xərcləri azaltmağa xidmət edir. Məqsəd funksiyaları bir-birinə ziddiyyət təşkil etdiyi üçün üsul, həmçinin işin cavabvermə vaxtını və tapşırığın maksimal emal vaxtını (*ing. makespan*) da azaldır. Təklif edilən üsulun çatışmazlığı onun tapşırıqların planlaşdırılması prosesinə çox vaxt sərf etməsi və sürətli olmamasıdır. [17]–də tapşırıqların yükünün heterogen resurslar arasında balanslaşdırılması üçün dinamik planlaşdırma alqoritmi təklif edilir. Yanaşmada bulud mühitində yük balanslaşdırılmasının təmin edilməsi məsələsi arıların qida mənbəyi axtarışındakı davranışı kimi modelləşdirilir. Bunun üçün VM-lər arasında yük balansını təmin etmək üçün arı alqoritmi istifadə edilir. Burada arılar balans tələb edən tapşırıqlar, VM-lər qida mənbəyi, az yüklənmiş maşınlar isə arıların təyinat yeri kimi modelləşdirilir. Burada VM yükləndiyi anda tapşırıq az yüklənmiş VM-ə yönləndirilir. Burada əsasən tapşırığın VM-i gözləməsinə sərf etdiyi vaxtı minimallaşdırmağa və ötürmə zolağını artırmağa çalışırlar. Üsulda emalda olan tapşırığın dayandırılmasına icazənin verilməməsinin (*ing. nonpreemptive*) əsas şərt kimi götürülməsi cavabvermə vaxtının və tapşırığın maksimal emal vaxtının azaldılmasına böyük imkan verir. Təklif edilən metodun üstünlüyü onun bulud mühitində yükü balanslaşdırmaqla yanaşı, VM növbəsində olan tapşırıqların sıra prioritetini nəzərə almasıdır. Yük balanslaşdırılması üçün bir-birindən asılı olmayan tapşırıqları miqrasiya etdirməsi və kifayət qədər miqyaslanan olmaması üsulun əsas çatışmazlığıdır. VM-lər arasında yük balanslaşdırılmasını təmin etmək üçün bal arılarının qidalanma davranışı kimi modelləşdirilən yanaşmalardan biri də [18]–də təklif edilmişdir. Üsulda yüklənmiş VM-lərdən götürülmüş tapşırıqlar bal arısı və az yüklənmiş VM-lər isə qida mənbəyi olduğu qəbul olunur. Təklif edilən üsul dörd blokdan ibarətdir. Birinci blokda VM-in cari iş yükü hesablanır, ikinci blokda yük balanslaşdırılması və planlaşdırılması üçün qərar qəbulu həyata keçirilir, üçüncü blokda VM-lərin qruplaşdırılması, dördüncü blokda tapşırığın planlaşdırılması həyata keçirilir. Burada əsasən tapşırığın maksimal emal vaxtının və VM miqrasiyalarının sayının azaldılmasına çalışırlar. Üsulun çatışmazlığı onun az miqyaslanan olmasıdır. Yük balanslaşdırılmasının effektiv təşkili, cavabvermə vaxtının optimallaşdırılması, tapşırığın yerinə yetirilmə müddətinin və maksimal emal vaxtının yaxşılaşdırılması, buraxma zolağında israfçılığın azaldılması bulud mühitində servis keyfiyyətinin (QoS) yüksəldilməsinə təsir edən kriteriyalardır. [19]–də yük balanslaşdırılmasını təmin etmək üçün yeni strategiyaya təklif olunur. Strategiyaya əsasən, tapşırıqlar siyahısından götürülmüş tapşırıqlar hesablama gücü yüksək olan VM-lərə yerləşdirilir. Üsulun məqsədi VM-in maksimal yükünü və tapşırığın yerinə yetirilmə müddətini optimallaşdırmaqla buludun QoS səviyyəsini yaxşılaşdırmaqdır. Üsulun çatışmazlığı onun miqyaslanan olmamasıdır. [4]–də emal xərclərini və ötürmə vaxtını minimallaşdırmaq məqsədi ilə kiçik mövqe qiymətinin (*ing. small position value, SPV*) üzərində qurulmuş PSO alqoritminə əsaslanan tapşırıqların planlaşdırılması üsulu təklif edilir. Üsul heterogen mühitlərdə olduqca yararlıdır. Üsulun çatışmazlığı miqyaslanmanı təmin etməməsidir. Yuxarıda icmal edilən dinamik yük balanslaşdırılması alqoritmlərinin xassələri və parametrləri cədvəl 1-də təsvir edilir.

Bulud texnologiyalarında yük balanslaşdırılması üçün olduqca çox sayda hibrid alqoritmlər də təklif edilmişdir [20, 21]. Təqdim edilən məqalədə dinamik yük balanslaşdırılması üçün hibrid yanaşmanın təklif edilməsi nəzərdə tutulmadığı üçün burada mövcud hibrid alqoritmlərin analizi aparılmır.

Dinamik yük balanslaşdırılması alqoritmləri və onların parametrləri

İstifadə olunan üsul	İdeya	İstinad	Üstünlüyü	Çatışmazlığı
SPV üzərində qurulmuş PSO alqoritmi	hesablama intensivliyi tələb edən tapşırıqların məhsuldarlığı yüksək olan kompüterə miqrasiya etdirilməsi	[4]	<ul style="list-style-type: none"> • emal xərclərinin minimallaşdırılması; • ötürmə vaxtının minimallaşdırılması; • heterogen sistemlərdə yararlı olması. 	miqyaslanma təmin edilmir
PSO	yük artıqlığı yaranan tapşırıqların uyğun VM-lərə yerləşdirilməsi	[14] [15]	<ul style="list-style-type: none"> • tapşırıqların emal vaxtının minimallaşdırılması; • tapşırıqların ötürülmə vaxtının minimallaşdırılması 	tapşırıqların bir-birindən asılı olmaması
genetik alqoritm	yük artıqlığı yaranan tapşırıqların uyğun VM-lərə yerləşdirilməsi	[16]	<ul style="list-style-type: none"> • tapşırıqların ötürülmə vaxtı; • tapşırıqların emal xərcləri; • enerji sərfiyyatı; • tapşırıqların növbəsinin uzunluğu. 	tapşırıqların planlaşdırılmasına çox vaxt sərf etməsi və sürətli olmaması
arı alqoritmi	arıların qidalanma davranışını modelləşdirir	[17]	<ul style="list-style-type: none"> • tapşırıqın VM-i gözləməsinə sərf etdiyi vaxtın minimallaşdırılması; • ötürmə zolağının artırılması; • VM növbəsində olan tapşırıqların sıra prioritetini nəzərə alması; • cavabvermə vaxtının azaldılması; • tapşırıqın maksimal emal vaxtının azaldılması 	<ul style="list-style-type: none"> • bir-birindən asılı olmayan tapşırıqları miqrasiya etdirməsi; • kifayət qədər miqyaslanan olmaması
Modifikasiya olunmuş bal arısı alqoritmi	bal arılarının qidalanma davranışını modelləşdirir	[18]	<ul style="list-style-type: none"> • cavabvermə vaxtının mini-mallaşdırılması; • tapşırıqın yerinə yetirilmə vaxtının minimallaşdırılması; • tapşırıqın maksimal emal vaxtının azaldılması; • VM miqrasiyalarının sayının azaldılması. 	az miqyaslanma
tapşırıqların yerləşdirilməsi üçün yeni strategiya	tapşırıqların yerləşdirilməsi strategiyasından istifadə etməklə yük balanslaşdırılmasının təmin edilməsi	[19]	<ul style="list-style-type: none"> • VM-in maksimal yükünün minimallaşdırılması; • tapşırıqın tamamlanma müddətinin azaldılması. 	miqyaslanan olmaması

Planlaşdırma məsələsinin qoyuluşu

Məqalədə iş axını təşkil edən tapşırıqların planlaşdırılması üçün üsul qurulur. İş axını adətən istiqamətlənmiş atsklik periodik olmayan qraf (*ing. Directed Acyclic Graph, DAG*) şəklində təsvir edilir və $G = (V, E)$ kimi yazılır [4]. Burada $V = \{T_1, T_2, \dots, T_n\}$ – iş axınındakı tapşırıqlar, n – iş axınındakı tapşırıqların ümumi sayıdır. $E = \{d_{ij}\}$, $1 \leq i, j \leq n$ qövslər çoxluğu, tapşırıqlar arasındakı verilənlərin asılılığını göstərir. Burada $d_{ij} = (T_i, T_j) \in E$ qövsü T_i tapşırıqından T_j tapşırıqına verilənlərin ötürüldüyünü göstərir.

Fərz edək ki, bulud mühitində m sayda VM vardır: $VM = \{VM_1, VM_2, \dots, VM_m\}$. Planlaşdırmanı həyata keçirmək üçün təklif edilmiş optimallaşdırma funksiyasına görə, tapşırıqların VM-lərə təyin edilməsi üçün optimal həllin tapılması tələb olunur. Burada tapşırıqın emal vaxtı (T_{exe}) və ötürülmə vaxtının (T_{trans}) minimallaşdırılması məqsəd funksiyaları kimi götürülmüşdür. Bu kriteriyalar aşağıdakı kimi hesablanır:

$$T_{exe} = \sum_{i=1}^n \sum_{k=1}^m x_{ik} * \frac{DE_i}{VM_{m_k}} = \frac{[\text{Əməliyyatların sayı}]}{\text{əməliyyatların sayı/san.}} = \text{san} \quad (1)$$

burada DE_i – i -ci tapşırıqın iş yükü (əməliyyatların sayı ilə göstərilir); VM_{m_k} – k -cı VM-dəki prosessorların sürəti (əməliyyat/san); m – VM-lərin sayı; n – tapşırıqların sayı.

$$T_{trans} = \sum_{i=1}^n \sum_{k=1}^m \sum_{z=1}^m (1 - x_{ik} * x_{iz}) \frac{DT_{kz}}{B_{kz}} \quad (2)$$

burada DT_{kz} – k -cı və z -ci virtual maşınlar arasında mübadilə edilən verilənlərin həcmi. i -ci tapşırıqın faylının (paketinin) həcminə bərabər götürülür (bit); B_{kz} , $k, z = \{1, 2, \dots, m\}$ – iki VM arasında ötürmə zolağı (bit/san), $x_{iz} = 1$ i -ci tapşırıq k -cı VM-dən götürülüb z -ci VM-ə təyin edilmişdir və belə olduqda, $x_{ik} = 0$ olur.

Mövcud metodların başlıca problemi kriteriyaların seçimindədir. Mövcud alqoritmlərdə planlaşdırma prosesində minimallaşdırılacaq kriteriyaların çəkisi eyni götürülür. Planlaşdırma prosesində kriteriyaların vaciblik dərəcəsini göstərmək üçün onlara müvafiq çəki əmsalları verilməlidir. Burada çəki əmsalları planlaşdırma prosesində kriteriyaların vacibliyini göstərmək üçün istifadə olunur. Kriteriyalara çəkilərin mənimsədilməsi daha yaxşı optimal həll tapmağa imkan verir. Bu münasibət aşağıdakı kimi ifadə olunur:

$$U = \alpha * T_{exe} + (1 - \alpha) * T_{trans} \rightarrow \min \quad (3)$$

burada α tapşırıqın emal vaxtına ($\alpha \in (0, 1)$), $(1 - \alpha)$ isə tapşırıqın ötürülmə vaxtına verilən çəki əmsalıdır. Burada α qiymətini dəyişməklə məqsəd funksiyasının daha optimal olmasını asanlıqla tənzimləmək mümkündür.

Məhdudiyət şərtləri

$$\sum_{k=1}^m x_{ik} = 1, \quad \forall i = 1, \dots, n \quad (4)$$

burada i -ci tapşırıq k -cı VM-ə təyin edildikdə, $x_{ik} = 1$, əks halda $x_{ik} = 0$.

$$\sum_{k=1}^m \sum_{z=1}^m (1 - x_{ik} * x_{iz}) = 1, \quad \forall i = 1, 2, \dots, n, \quad k \neq z \quad (5)$$

$$x_{ik}, (1 - x_{ik} * x_{iz}) \in \{0, 1\}, \forall i, k, z$$

burada i -ci tapşırıq k -cı VM-dən götürülüb, z -ci VM-ə təyin edildikdə, $x_{iz} = 1$, əks halda $x_{iz} = 0$.

Üsulda aşağıdakı dəyişənlər də istifadə edilmişdir:

$T_{set} = \{T_1, T_2, \dots, T_n\}$ – miqrasiya olunacaq tapşırıqlar çoxluğu;

VM_j – j -cu virtual maşın, $j = \{1, 2, \dots, m\}$;

MIPS (*ing. million instructions per second*) – saniyədə milyon əməliyyat.

PSO alqoritminin kanonik modeli

PSO alqoritminin kanonik modeli hissəciklər sürüsündən ibarətdir. Bu hissəciklər hiper-ölçülü axtarış fəzasında “uçur”. Hissəciklərin axtarış fəzasında mövqe dəyişmələri fərdlərin sosial-psixoloji meylinə əsaslanır. Burada hər bir hissəciyin mövqeyi onun özünün və qonşularının səriştəsinə əsasən dəyişir.

PSO alqoritmində hissəciklər namizəd həllər kimi təqdim edilir. Hissəciklər məqsəd funksiyasının ən yaxşı qiymət aldığı həlli tapmaq üçün iterativ addımlar həyata keçirir. Hər bir hissəcik \vec{x}_i (i – hissəciyin indeksidir) mövqe vektoru vasitəsi ilə təsvir edilən mövqeyə, və \vec{v}_i sürət vektoru ilə təsvir edilən sürətə malikdir. İterativ prosedə hər bir hissəcik özünün ən yaxşı (*ing. best position*) mövqeyini \vec{x}_{pbest_i} vektorunda yadda saxlayır. Sürünün arasındakı ən yaxşı mövqe vektoru isə \vec{x}_{gbest_i} vektorunda saxlanır. Hər bir t iterasiyasında sürət vektorunun yenilənməsi (6) düsturu vasitəsi ilə həyata keçirilir. Daha sonra isə (7) düsturu əsasında əvvəlki mövqe ilə yeni sürəti toplamaqla yeni mövqe formalaşdırılır.

$$\vec{V}_i(t+1) = W\vec{V}_i(t) + C_1r_1(\vec{x}_{pbest_i} - \vec{X}_i(t)) + C_2r_2(\vec{x}_{gbest_i} - \vec{X}_i(t)) \quad (6)$$

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{V}_i(t+1) \quad (7)$$

burada, W – inersiya faktorudur ($W = 0.95$), C_1 – müsbət sabit ədəddir və özünü tanıma əmsalı kimi elan edilir ($C_1 = 0.8$), C_2 – müsbət sabit ədəddir və sosial komponent əmsalı kimi elan edilir ($C_2 = 0.8$). Burada $C_1 + C_2 \leq 4$ olmalıdır. r_1 və r_2 təsadüfi ədədlərdir, populyasiyanın fərqliliyini göstərmək üçün istifadə edilir və $[0, 1]$ intervalında qiymətlər alır.

Burada hissəcik (6) düsturuna əsasən özünün mövqeyinə və sürüdəki ən uğurlu hissəciyin mövqeyinə əsasən növbəti dəfə hansı mövqeyə hərəkət edəcəyini təyin edir.

Hissəciklərin PSO vasitəsilə (6) və (7) düsturlarını tətbiq etməklə yaradılan bütün mövqeləri müəyyən bir vektor əmələ gətirir. Bu vektorun qiymətləri təbiətinə görə kəsilməzdir [22] və bu kəsilməz ədədlərlə VM-lərin nömrəsini müəyyən etmək qeyri-mümkün hesab olunduğundan, onların diskret ədədlərə çevrilməsi vacib hesab olunur [22, 23].

Hissəciyin mövqeyinin kəsilməz qiymətlərdən ibarət $x_i^k = [x_1^k, x_2^k, \dots, x_n^k]$ vektorunu diskret vektora çevirmək üçün kiçik mövqe qiyməti qaydasından istifadə edirlər [4, 24].

Kiçik Mövqe Qiyməti

Kəsilməz $x_i^k = [x_1^k, x_2^k, \dots, x_n^k]$ mövqe vektorunu diskret ədədlərdən ibarət $s_i^k = [s_1^k, s_2^k, \dots, s_n^k]$ vektoruna çevirmək üçün SPV qaydası aşağıdakı kimidir [4, 24]:

SPV qaydasına görə, x_i^k həllər sırasında ən kiçik PSO qiymətinə malik tapşırıqdan başlayaraq bütün tapşırıqlar artma ardıcılığına görə nömrələnir və s_i^k xanasına daxil edilir (cədvəl 2). Burada i – i -ci hissəcik (PSO həlli), k – k -cı iterasiyadır.

Cədvəl 2

Kəsilməz PSO ədədlərinin SPV qaydası vasitəsilə diskretləşdirilməsi

Tapşırıqlar	1	2	3	4	5	6	7	8	9	10
x_i^k	0.1587	3.6189	2.3824	0.0292	0.8254	0.4256	0.3679	0.1276	0.6378	0.7832
s_i^k	3	10	9	1	8	5	4	2	6	7

Cədvəl 2-dən görüldüyü kimi, SPV çevirməsindən sonra tapşırıqlar üçün alınan plan 3–10–9–1–8–5–4–2–6–7 olacaq.

SPV qiymətləri formalaşdırıldıqdan sonra aşağıdakı düstur vasitəsi ilə s_i^k çevirmə vektorunun elementləri uyğun VM vektoruna inikas etdirilir və beləliklə, planlaşdırma vektoru formalaşdırılır:

$$p_i^k = s_i^k \text{ mod } m + 1 \quad (8)$$

burada m virtual maşınların ümumi sayını göstərir. Bizim halda 5 VM götürülmüşdür ($VM_i^k = 1, 2, 3, 4, 5$). x_i^k – hissəciklərin PSO vasitəsilə alınmış kəsilməz qiymətlərdən ibarət vektorudur. s_i^k – kəsilməz x_i^k qiymətlərinin SPV çevirməsi nəticəsində alınmış diskret qiymətlərdir. (8) düsturu vasitəsilə diskret s_i^k qiymətlərinin (tapşırıqlarının) VM-ə inikası aşağıdakı kimi olacaq:

$$\begin{aligned} p_1 &= 3 \text{ mod } 5 + 1 = 3 + 1 = 4; \\ p_2 &= 10 \text{ mod } 5 + 1 = 0 + 1 = 1; \\ p_3 &= 9 \text{ mod } 5 + 1 = 4 + 1 = 5; \\ p_4 &= 1 \text{ mod } 5 + 1 = 1 + 1 = 2; \\ p_5 &= 8 \text{ mod } 5 + 1 = 3 + 1 = 4; \\ p_6 &= 5 \text{ mod } 5 + 1 = 0 + 1 = 1; \\ p_7 &= 4 \text{ mod } 5 + 1 = 4 + 1 = 5; \\ p_8 &= 2 \text{ mod } 5 + 1 = 2 + 1 = 3; \\ p_9 &= 6 \text{ mod } 5 + 1 = 1 + 1 = 2; \\ p_{10} &= 7 \text{ mod } 5 + 1 = 2 + 1 = 3. \end{aligned}$$

Cədvəl 3

PSO alqoritminin kəsilməz qiymətlərinin 10 tapşırıq və 5 VM əsasında inikası

Tapşırıqların sayı	x_i^k	s_i^k	VM-lərə inikası (p_i^k)
1	0.1587	3	4
2	3.6189	10	1
3	2.3824	9	5
4	0.0292	1	2
5	0.8254	8	4
6	0.4256	5	1
7	0.3679	4	5
8	0.1276	2	3
9	0.6378	6	2
10	0.7832	7	3

Yük balanslaşdırılması üçün PSO parametrləri

Təklif edilmiş üsul VM-lər arasında effektiv yük balanslaşdırılmasını təmin etmək üçün PSO alqoritmindən istifadə edir və yük artıqlığı yaradan tapşırıqları az yüklənmiş VM-lərə yerləşdirir. PSO alqoritminin parametrlərinin bulud mühitinə inikası cədvəl 4-də verilir.

Cədvəl 4

PSO alqoritminin parametrlərinin bulud mühitinə inikası

PSO parametrləri	Bulud mühiti
Hissəcik (<i>ing. particle</i>)	Tapşırıqlar
Hissəciklərin mövqeləri	VM-in sıra nömrəsi
Optimal həll	N sayda tapşırıqdan ibarət n ölçülü vektordur. Məsələn, tapşırıq (<i>ing. cloudlet</i>) planlarından biri. Bu vektorun elementləri tapşırıqlardır və $[1, n]$ arasında qiymətlər alır.
Hissəciyin miqrasiyası	Tapşırıqın gücü çox olan az yüklənmiş VM-lərə yerləşdirilməsi

Təklif edilən üsulda tapşırıqlar hissəciklərdir. Burada hissəciklərin “qida mənbəyi” axtarışına çıxdığı ilkin mərhələsində, tapşırıqlar emal olunmaq üçün VM-lərə təyin edilir. Ayrı-ayrı VM-lərin emal imkanları fərqli olduğundan, bəzən bu VM-lər həddindən artıq yüklənməyə məruz qalır, digərləri isə az yüklənmiş olur. Uyğun VM çoxsaylı tapşırıqlarla yüklənmiş olduqda, burada bəzi tapşırıqlar az yüklənmiş VM-lərə miqrasiya etdirilir.

Eksperiment

Təklif edilmiş üsulun imkanlarını qiymətləndirmək üçün Jswarm və Cloudsim proqramlarından istifadə edilmişdir. Cloudsim proqramının “DatacenterBroker” sinifinə daxil olan “bindCloudletToVm()” brokeri tapşırıqların VM-lərə təyin edilməsini həyata keçirir. Jswarm isə tapşırıqların VM-lər arasında optimal düzümünü PSO alqoritminə əsasən tapır. Təqdim olunan məqalədə təklif edilən α PSO – TBLB modelində “bindCloudletToVm()” brokeri tapşırıqları VM-lərə Jswarm alqoritminin nəticələrinə əsasən təyin edir.

Cloudsim proqramında bulud simulyasiya mühitini yaratmaq üçün üç fiziki maşın (verilənlər mərkəzi), beş VM, on tapşırıq götürülmüşdür. α PSO – TBLB modelində VM-lər və tapşırıqlar üçün istifadə olunan verilənlər Cədvəl 5 və Cədvəl 6-da verilmişdir.

Cədvəl 5

VM-lərin xassələri

VM-in nömrəsi	Sürət (MIPS)	VM təsvirinin ölçüsü	VM-in yaddaşı (ram)	Ötürmə zolağı (<i>ing.bandwidth</i>)	Processorların (CPU) sayı	Virtual maşın monitorunun adı (VMM)
1	250	1,000	512	1,000	1	Xen
2	300	1,000	256	1,000	1	Xen
3	250	1,000	512	1,000	1	Xen
4	250	1,000	512	1,000	1	Xen
5	250	1,000	512	1,000	1	Xen

Təklif edilmiş üsul əsasında yük balanslaşdırılmasını təmin etmək üçün fərz edilir ki, beş münasib VM var. Belə olduğu halda, növbəti addımda Cloudsim proqramından istifadə etməklə yüklənmiş VM-də yük artıqlığı yaradan tapşırıqların münasib VM-lərə tapşırığın emal vaxtını və tapşırığın ötürülmə vaxtını minimallaşdırmaqla optimal təyin edilməsi həyata keçirilir.

Cədvəl 6

Tapşırıqların xassələri

Tapşırığın nömrəsi	Tapşırığın əməliyyatlarının sayı	Tapşırığın faylının həcmi	Tapşırığın çıxış verilənlərinin həcmi	Processorların (CPU) sayı
1	250,000	300	300	1
2	25,000	300	300	1
3	250,000	300	300	1
4	25,000	300	300	1
5	250,000	300	300	1
6	250,000	300	300	1
7	25,000	300	300	1
8	250,000	300	300	1
9	250,000	300	300	1
10	25,000	300	300	1

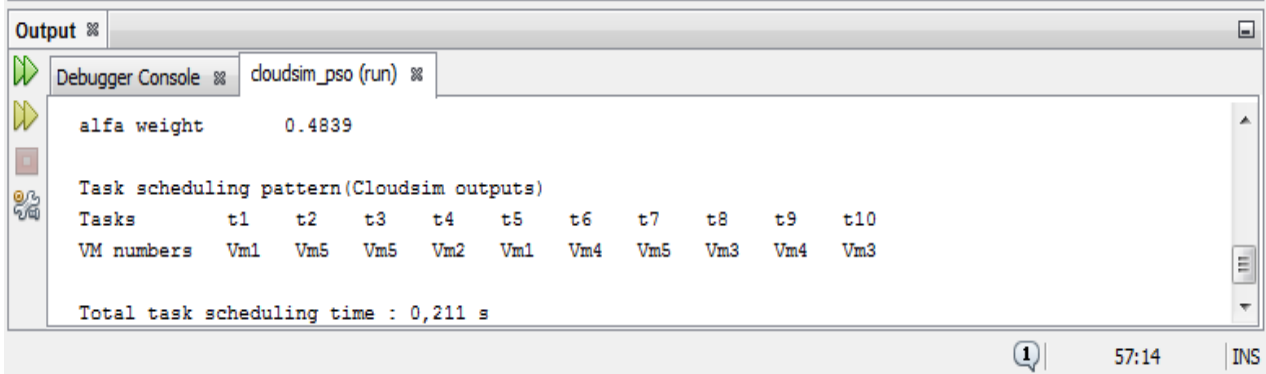
Bu tapşırıqların VM-lər üzərində planlaşdırılması üçün optimal planın tapılması təklif edilmiş α PSO – TBLB alqoritmi əsasında həyata keçirilir. Daha sonra bu optimal plan Cloudsim proqramındakı “bindCloudletToVm ()” brokeri tərəfindən tapşırıqların optimal planlaşdırılması üçün istifadə edilmişdir.

[3] PSO alqoritmini tətbiq etməklə əlavə tapşırıqların yeni host VM-lərə miqrasiyasını təmin edən üsul təklif edir. Burada bizim təklif etdiyimiz üsula yaxın yanaşma irəli sürülür. Lakin bu yanaşmada aparılan eksperimentlərdən aydın şəkildə görünür ki, burada yükün paylanmasında ciddi disbalans vardır. Burada, VM1 maşınına iki – t_5 və t_6 , VM3 maşınına dörd – t_2 , t_4 , t_7 və t_8 , VM4 maşınına iki – t_1 və t_{10} , VM5 maşınına iki – t_3 və t_9 tapşırıqları təyin edilmişdir, VM2 isə emal prosesində iştirak etmir. Lakin α PSO – TBLB alqoritmi əsasında Cloudsim-də alınmış nəticədən (Cədvəl 7) görünür ki, bizim təqdim etdiyimiz yanaşmada bu disbalans α çəki əmsalını tətbiq etməklə ciddi şəkildə aradan qaldırılır. Belə ki, burada çəki əmsalının $\alpha = 0.4839$ qiymətində tapşırıqlar VM-lər arasında bərabər paylanır və tapşırıqların emalı prosesində bütün VM-lər iştirak edir. Cədvəl 7-dən görüldüyü kimi, VM1 maşınına iki – t_1 və t_5 , VM2 maşınına bir – t_4 , VM3 maşınına iki – t_8 , t_{10} , VM4 maşınına iki – t_6 və t_9 , VM5 maşınına üç – t_2 , t_3 və t_7 tapşırıqları yüklənmişdir.

Simulyasiyanın nəticələri

Tapşırıqlar	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
VM-lər	vm_1	vm_5	vm_5	vm_2	vm_1	vm_4	vm_5	vm_3	vm_4	vm_3
Tapşırığın ümumi planlaşdırma vaxtı: 0.211 saniyə										

Bundan əlavə α PSO – TBLB metodunda tapşırıqların planlaşdırılmasına sərf olunan ümumi vaxtın optimallaşdırılması da çox uğurla aparılmışdır. Belə ki, [3]-də bu vaxt müddəti 0.224 saniyə təşkil etdiyi halda, α PSO – TBLB metodunda bu göstərici 0.211 saniyə təkil etmişdir (şəkil 1).



Şəkil 1. Tapşırıqların Cloudsim-də planlaşdırılması

Hissəciklərin şəkil 1-də təsvir edilmiş mövqeləri PSO alqoritminə görə optimal həldir: $d(\bar{X}_i) = (d_1, d_2, \dots, d_{10}) = (1, 5, 5, 2, 1, 4, 5, 3, 4, 3)$. Bu optimal həllə nəzərən $vm_1, vm_5, vm_5, vm_2, vm_1, vm_4, vm_5, vm_3, vm_4, vm_3$ virtual maşınları uyğun olaraq $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}$ tapşırıqlarını emal etmək üçün seçilmişdir.

Bulud texnologiyaları mühitində ənənəvi yük balanslaşdırılmasını təmin etmək üçün ilk öncə yeni VM-in yaradılmasına zərurət yaranır. Bu mühitdə VM-in yaradılması prosesinə təxminən 5-15 dəqiqə sərf olunur. Lakin yük balanslaşdırılmasının təklif edilmiş α PSO – TBLB modelinə əsasən təmin edilməsi prosesinə 0.211 saniyə sərf olunur. Bu işə ənənəvi yük balanslaşdırılması yanaşması ilə müqayisədə kifayət qədər az vaxtın sərf olunması ilə nəticələnir.

Planlaşdırmanın həyata keçirilməsindən sonra gələn proses miqrasiya prosesidir. Ənənəvi yük balanslaşdırılması yanaşmalarında ilkin VM-in yeni host VM-ə bütünlükdə miqrasiya etdirilməsi lazım gəlir. Lakin təklif edilmiş α PSO – TBLB metodunda yalnız yük artıqlığı yarıdan tapşırıqların münasib VM-lərə miqrasiya etdirilməsi həyata keçirilir. Aydın ki, VM-in bütünlükdə miqrasiyası olduqca çox vaxtın və yaddaşın sərf olunması ilə nəticələnir. Lakin yalnız tapşırıqların miqrasiyası prosesinə saniyələrlə vaxt sərf olunur və yaddaş yüklənməsinin, demək olar ki, qarşısı alınır.

Nəticə

Bulud texnologiyaları çoxkriteriyalı bir mühitdir. Bu mühitdə bulud resurslarının bir kriteriya əsasında planlaşdırılmasının təşkili VM-lər arasında ciddi yük disbalansının yaranmasına səbəb olur. Bu səbəbdən təqdim olunan məqalədə bu problemi həll etmək üçün tapşırıqların çəkili planlaşdırılmasını həyata keçirən yeni çoxkriteriyalı optimallaşdırma metodu təklif edilir. Yük balanslaşdırılması üçün tapşırıqların planlaşdırılması sahəsində irəli sürülmüş mövcud metodların başlıca problemi kriteriyaların seçimindədir. Mövcud alqoritmlərdə planlaşdırma prosesində minimallaşdırılacaq kriteriyaların çəkisi eyni götürülür. Planlaşdırma prosesində kriteriyaların vacibliyini göstərmək üçün onlara müvafiq çəki əmsalları verilməlidir. Bu çəki əmsalları planlaşdırma prosesində kriteriyaların vacibliyini göstərmək üçün istifadə olunur. Kriteriyalara çəkilərin mənimsədilməsi daha optimal həll tapmağa imkan verir. Çəki əmsallarının qiymətini dəyişməklə məqsəd funksiyasının daha optimal olmasını asanlıqla tənzimləmək mümkündür.

Ədəbiyyat

1. Metri G., Srinivasaraghavan S., ShiW., Brockmeyer M. Experimental analysis of application specific energy efficiency of datacenters with heterogeneous servers / Proc. of the IEEE 5th International Conference on Cloud Computing, 2012, pp.786–793.
2. Vaquero L.M., Rodero-Merino L., Caceres J., Lindner M. A break in the clouds: towards a cloud definition // ACM SIGCOMM Computer Communication Review, 2008, vol.39, no.1, pp.50–55.
3. Ramezani F., Lu J., Hussain F.K. Task-based system load balancing in cloud computing using Particle Swarm Optimization // International Journal of Parallel Programming, 2013, vol.42, no.5, pp.739–754.
4. Guo L., Zhao S., Shen S., Jiang C. Task scheduling optimization in cloud computing based on heuristic algorithm // Journal of Networks, 2012. vol.7, no3, pp.547–553.
5. Alguliev R.M., Alyguliev R.M., Alekperov R.K. An approach to optimal task assignment in a distributed system // Journal of Automation and Information Sciences, 2004, vol.36, no.10, pp.51–55.
6. Танаев В.С., Гордон В.С., Шафранский Я.М. Теория расписаний. Одностадийные системы. М.:Наука. Главная редакция физико-математической литературы, 1984, с.384.
7. Wu Z., Liu X., Ni Z., Yuan D., Yang Y. A market-oriented hierarchical scheduling strategy in cloud workflow systems // The Journal of Supercomputing, 2013, vol.63, no.1, pp.256–293.
8. Ajit M., Vidya G. VM level load balancing in cloud environment / Proc. of the fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp.87–95.
9. Chawla Y., Bhonsle M. A study on scheduling methods in cloud computing // International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2012, vol.1, no.3, pp.12–17.
10. Mohammad M., ValiKardan S., Shahi Z., Azar S.I. Towards workflow scheduling in cloud computing: A comprehensive analysis // Journal of Network and Computer Applications, 2016, vol.66, pp. 64–82.
11. Milani A.S., Navimipour N.J. Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends // Journal of Network and Computer Applications, 2016, vol.71, pp.86–98.
12. Ajit M., Vidya G. VM level load balancing in cloud environment / Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp.1–5.
13. Madni S.H., Latiff M.S., Coulibaly Y., Abdulhamid S.M. Resource scheduling for Infrastructure as a Service (IaaS) in cloud computing: Challenges and opportunities // Journal of Network and Computer Applications, 2016, vol.68, pp.173–200.
14. Ramezani F., Lu J., Hussain F. Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization // Service-Oriented Computing, 2013, vol.8274, pp.237–251.
15. Ramezani F., Lu J., Hussain F.K. Task-based system load balancing in cloud computing using particle swarm optimization // Knowledge Engineering and Management, 2013, pp.31–42.
16. Ramezani F. Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments // World Wide Web, 2015, vol.18, no.6, pp.1737–1757.
17. Dhinesh B.D., Krishna P.V. Honey bee behavior inspired load balancing of tasks in cloud computing environments // Applied Soft Computing, 2013, vol.13, no.5, pp.2292–2303.
18. Babu K.R., Samuel P. Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud // Innovations in Bio-inspired Computing and Applications, 2016, pp.67–78.
19. Banerjee S., Adhikari M., Kar S., Biswas U. Development and analysis of a new cloudlet allocation strategy for QoS improvement in cloud // Arabian Journal for Science and Engineering, 2015, vol.40, no.5, pp.1409–1425.
20. Liu Y., Zhang C., Li B., Niu J. DeMS: A hybrid scheme of task scheduling and load balancing

- in computing clusters // Journal of Network and Computer Applications, 2015, pp.1–8.
21. Cho K.M., Tsai P.W., Tsai C.W., Yang C.S. A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing // Neural Computing and Applications, 2015, vol.26, no.6, pp.1297–1309.
 22. Alguliev R.M., Aliguliyev R.M., Mehdiyev C.A. An optimization approach to automatic generic document summarization // Computational Intelligence, 2013, vol.29, no.1, pp.129–155.
 23. Aliguliyev R.M. Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization // Computational Intelligence, 2010, vol.26, no.4, pp.420–448.
 24. Cakar T., Koker R. Solving Single Machine Total Weighted Tardiness Problem with Unequal Release Date Using Neurohybrid Particle Swarm Optimization Approach // Computational Intelligence and Neuroscience, 2015, vol.2015, pp.1–13.

УДК 004.451.44

Алгулиев Расим М.¹, Имамвердиев Ядигар.Н.², Абдуллаева Фаргана Д.³

Институт Информационных Технологий НАНА, Баку, Азербайджан

¹rasim@science.az, ²yadigar@iit.science.az, ³farqana@iit.ab.az

Многокритериальный оптимизационный метод для балансировки нагрузки в облачных технологиях

Оптимизация процесса планирования задач в облачной среде является многокритериальной NP-сложной задачей. В статье для балансировки нагрузки был предложен весовой метод PSO–TBLB, основанный на алгоритме PSO. Метод обеспечивает оптимальную миграцию задач, создающих избыточную нагрузку на виртуальные машины инфраструктуры облачных технологий из загруженных виртуальных машин в менее загруженную виртуальную машину. В предложенном оптимизационном методе минимизация времени обработки задач и времени передачи заданий были выбраны в качестве целевых функций. Экспериментальная проверка предложенного подхода была проведена в программах Jswarm и Cloudsim. В результате симуляции на основе предложенного метода найдено оптимальное решение для планирования задач, было обеспечено равномерное распределение задач в виртуальные машины и было достигнуто потребление малого объема времени в процессе назначения задач в виртуальные машины.

Ключевые слова: облачные технологии, метод роя частиц, миграция виртуальных машин, планирование задач, Cloudsim, Jswarm, интенсивность данных, интенсивность вычислений.

Rasim M. Alguliyev¹, Yadigar N. İmamverdiyev², Fargana C. Abdullayeva³

Institute of Information Technology of ANAS, Baku, Azerbaijan

¹rasim@science.az, ²yadigar@iit.science.az, ³farqana@iit.ab.az

Multicriteria optimization method for load balancing in cloud technologies

Optimizing of the task scheduling process in the cloud environment is a multicriteria NP-hard problem. In this paper weighted load balancing method (α PSO – TBLB) based on PSO algorithm is proposed. The method provides optimal migration of tasks from the loaded virtual machines to the less loaded virtual machine to prevent the excessive load in virtual machines of the infrastructure of cloud technologies. In the proposed optimization method, the minimization of the processing time of tasks and the transfer time of tasks were selected as the objective functions. Experimental testing of the proposed approach was carried out in the Jswarm and Cloudsim programs. As a result of the simulation on the basis of the proposed method, an optimal solution for task scheduling was found, uniform distribution of tasks in virtual machines was provided. Moreover, in the process of assigning tasks to virtual machines, a minimal time consumption was achieved.

Keywords: cloud computing, Particle Swarm Optimization (PSO), virtual machine migration, task scheduling, Cloudsim, Jswarm, data intensive, computing intensive.