

anında daha gelişmiş planlama yapabilmek için hibrid sistemler kullandığı da söylenebilir.

2.3 Dağıtık Sistemlerde Dinamik Yük Dengeleme

2.3.1 Yük dağıtım gereksinimi

Dağıtık sistemlerin güçlü işlem kapasitesinin avantajlarından etkin bir şekilde yararlanabilmek için iyi bir kaynak dağıtım modeline ihtiyaç bulunmaktadır. Dağıtık işlem programlayıcısı (distributed process scheduler), sistemin genel performansını en üst seviyede tutacak şekilde sistemin yükünü adil ve saydam bir şekilde işlemciler arasında dağıtır (Singhal and Shivaratri, 1994). Geniş alan ağlarındaki yüksek iletişim gecikmelerinden dolayı dağıtık işlem dağıtım yerel ağlarda çalışan dağıtık sistemler için daha uygundur.

Dağıtık yük dağıtımına duyulan ihtiyacı açıklamak için şöyle bir senaryo düşünülebilir: Yerel bir ağ ile birbirine bağlı durumdaki bağımsız bilgisayarlardan oluşan dağıtık bir sistemde kullanıcılar kendi bilgisayarlarından sisteme birtakım işler (task) yüklerler. Böyle bir sistemde, sisteme farklı zaman dilimlerinde yüklenen farklı servis zamanlarına sahip işlemler söz konusu olduğunda yüksek bir olasılıkla bazı bilgisayarlar ağır iş yükü altında ezilirken, bazıları boş durumda olacaklardır. Eğer bazı bilgisayarlara yüklenen iş yoğunluğu diğerlerinden fazlaysa, ya da bazı işlemciler diğerlerinden daha yavaş işlem hızına sahiplerse bu durumun daha da sıklıkla oluşacağı açıktır. Tüm bilgisayarların eşit işlem kapasitesine sahip olduğu ve uzun dönem iş yüklerinin birbirlerine eşit yoğunlukta olduğu bir sistemde ilk bakışta

yük dağıtımının gereksiz olduğu düşünülse de, yapılan bazı çalışmalarda bu tür homojen dağıtık sistemlerde bile iş geliş ve servis zamanlarındaki değişimlerin yüksek olasılıkla, belli bir anda sistemde bir iş servis beklerken en az bir işlemcinin boş durumda olduğu ortaya çıkarılmıştır (Livny and Melman, 1982).

Sonuç olarak, dağıtık sistemlerde, sistem homojen bir yapıda olsa bile işlerin, yükü ağır işlemcilerden hafif yüklü işlemcilere transferi sistemin genel performansını arttırmaktadır. Bu noktada sistem performansının nasıl belirleneceği sorusu akla gelebilir. Sistem performans ölçütlerinden biri işlemlere verilen ortalama cevap verme süresi (average response time), yani bir işin sisteme yüklenip, sonucunun geri dönmeye kadar geçen süredir. Yük dağıtımının amacı sıklıkla bu süreyi düşürmektir. Akla gelebilecek bir başka soru ise bir işlemcideki yük durumunun nasıl belirleneceğidir. Bu konudaki ölçütlerden sıklıkla kullanılan işlemcinin iş kuyruğunun boyudur (CPU queue length).

Sistem performansını arttırmak bir yük dağıtım modelinin ana hedefi olmakla birlikte, karşılaması gereken birkaç önemli gereksinim daha bulunmaktadır (Singhal and Shivaratri, 1994):

Ölçeklenebilirlik: Yük dağıtım modeli, daha büyük dağıtık sistemlerde de çalışabilmelidir. Bu, minimum maliyetle hızlı dağıtım kararları alabilmeyi gerektirir.

Lokasyon bağımsızlığı: Transfer edilen işlem transfer edildiği işlemcide de orjinal işlemcide çalıştırılması halinde üretilen sonuçların aynı üretilmelidir.

İşi bölme (preemption) : Bir işstasyonunun sahibi yokken, kaynaklarından yararlanmak için yük transferi yapılırken, o işstasyonunun sahibi bilgisayarını kullanmak istediğinde bu kişiye düşük bir performans sunulmamalıdır. Bunun için transfer edilmiş işler durdurulmalı ve iş istasyonu sahibinin işlerine öncelik verilmelidir.

Heterojenlik: Yük dağıtım modeli, farklı mimarilerde, değişik donanımlarda çalışan sistemleri ayırt edebilmelidir.

2.3.2 Yük dağıtımı ile ilgili karakteristikler

2.3.2.1 Yük değeri ölçümü

İşlemci kuyruk uzunluğu, işlem cevap verme süresini doğrudan etkileyen bir birim olduğu için yük değeri olarak iyi bir belirteçdir. Üstelik işlemci kuyruk uzunluğunu ölçmek kolay ve sisteme çok az yük getiren bir işlemdir. Bununla birlikte eğer yük transferi belli bir gecikmeye sebep olabiliyorsa, yük miktarı için sadece kuyruk uzunluğunu kullanmak bir işlemcinin bir yük transferi henüz tamamlanmamışken, diğer istekleri de kabul etmesine neden olabilir. Bu durumda diğer kabul ettiği yük transferleri de tamamlandığında bu işlemci fazla iş yükü altında kalabilir. Bunu önlemenin bir yolu, işlemci yük transferini kabul ettikten sonra yük değerini manuel olarak arttırmaktır. Eğer transfer başarısızlıkla sonuçlanırsa, yani belli bir sürede gerçekleşmezse, yük değeri tekrar eski değerine alınır.

Yapılan çalışmalar işlemci kuyruk uzunluğunun işlemci kullanılabilirliği (utilization) üzerinde özellikle interaktif sistemlerde az

bir etkisi olduğunu göstermiştir. Bu nedenle bazı sistemler yük değeri ölçümünde işlemci kullanılabilirliğini kullanmışlardır. Fakat kullanılabilirliği ölçmek için arkaplanda çalışan işlemler kullanılmakta, bu da sisteme ek yük getirmektedir.

2.3.2.2Yük dağıtım algoritmaları sınıflandırması

Bir yük dağıtım algoritmasının en temel fonksiyonu yükü ağır makinelerden yükü hafif makinelere iş transferi yapmaktır. Yük dağıtım algoritmaları genel olarak üç gruba ayrılabilirler: Statik, dinamik ve adaptif. Dinamik yük dağıtım algoritmaları sistem durum bilgilerini (işlemcilerdeki yük miktarı) kullanarak yükün dağıtımını ile ilgili kararlar verirler. Statik algoritmalar ise sistem durum bilgisini kullanmaksızın, sistem hakkında önceden edinilmiş ve algoritmanın içine gömülmüş birtakım kurallara göre dağıtım yaparlar. Dinamik algoritmalar sistemin yük durumunda kısa süreli değişimler yaratarak statik algoritmalara göre daha iyi performans sağlayabilirler. Bunun yanında dinamik algoritmalar sistem durum bilgisi toplama, işleme ve analiz etme gibi işlemler gerektirmesi dolayısı ile sisteme ek yük getirirler. Adaptif yük dağıtım algoritmaları, sistem durumuna göre algoritmanın bazı parametrelerini değiştirerek kendi aktivitelerini ayarlayan, dinamik algoritmaların bir başka çeşididir. Örnek vermek gerekirse dinamik bir yük dağıtım algoritması sistemin genel yük durumu yüksek olduğunda bile, sistem durum bilgisi toplamaya devam edecektir. Bu durumda sistemde yükü hafif bir işlemci bulma olasılığı çok düşük olduğundan hem gereksiz yere işlem yapacak, hem de sistemin zaten fazla olan yükünü biraz daha arttıracaktır. Adaptif bir algoritma ise böyle bir durumda gereksiz bilgi toplama işlemini durduracaktır.

2.3.2.3Yük dengelemesi ve yük paylaşımı

Yük dağıtım algoritmaları dağıtım prensiplerine göre, yük dengelemesi ve yük paylaşımı algoritmaları olarak farklı şekilde de sınıflandırılabilirler. Her iki çeşit algoritma da sistemin yük durumu açısından paylaşımsız durumunu (bazı işlemciler çok yüklüken, bazılarının boş olduğu durum) azaltmaya çalışırlar. Fakat yük dengeleme algoritmaları bunun bir adım daha önüne geçerek tüm işlemcilerdeki yük seviyesini eşitlemeye çalışırlar. Bu amaçları sonucu yük dengeleme algoritmaları yük paylaşım algoritmalarına göre daha yüksek bir yük transferi yoğunluğuna sahiptirler ve bu da sisteme daha fazla yük getirerek bazı durumlarda sistem performansına olumsuz etkide bulunurlar.

Yük transferleri iletişim gecikmeleri ve iş durumlarını toplama sırasında yaşanan gecikmeler dolayısı ile anlık olarak gerçekleşmez. Yük transferindeki gecikmeler sistemin paylaşımsız durumunun süresini de uzatır. Bunu önlemenin bir yolu kısa bir süre içerisinde boş duruma düşebilecek işlemcilere tahmini yük transferi yapmaktır. Bu tür tahmini transferler yük transfer yoğunluğunu arttıracığından bu tür yük paylaşımı algoritmaları yük dengeleme algoritmalarına benzer bir haldedirler. Bu nedenle yük dengeleme algoritmalarına yük paylaşımı algoritmalarının tahmini yük transferi yapan özel bir şeklidir diyebiliriz.

2.3.2.4Bölerek (preemptive) veya bölmeden (nonpreemptive)

Bölünmüş yük transferleri, işletilmeye başlanmış işlerin çalışmasının kesilerek bir başka işlemciye transfer edilmesini ifade eder. Bu transfer, kesintiye uğratılan işin büyük ve karmaşık olabilecek tüm durum bilgilerinin toplanarak işle birlikte transfer edilmesini gerektirdiğinden oldukça maliyetli ve zor bir işlemdir. Buna karşılık bölünmemiş yük transferleri yalnızca henüz işletilmeye başlamamış, dolayısı ile durum bilgisi bulunmayan işleri transfer ederler. Her iki durumda da işle ilgisi çevre bilgilerinin (çalışma klasörü, işin miras aldığı erişim hakları, vb.) transfer edilmesi gerekmektedir. Bölünmemiş yük transferleri aynı zamanda yük yerleştirmesi şeklinde de adlandırılırlar.

2.3.2.5Dengelilik (stability)

Bir sistemin uzun dönem iş geliş oranı, sistemin iş yapabilme kapasitesinden fazlaysa, işlemci kuyrukları sürekli bir şekilde büyümeye başlar. Böyle bir sistem dengesiz (unstable) olarak ifade edilir. Sistem durum bilgisi toplamak için sürekli mesaj iletişimi yapan bir yük dağıtım algoritması düşünelim. Sisteme yüklenen iş miktarı ile bu algoritma tarafından sisteme getirilen yük miktarı birlikte sistemin servis kapasitesini aşabilir ve sistemi dengesiz bir hale sürükleyebilir.

Bunun yanısıra, bir algoritma dengeli bir yapıda olsa bile sistemi yük dağıtımının hiç yapılmadığı durumdan daha kötü bir performansa sürükleyebilir. Bu nedenle algoritmaları daha kısıtlayıcı bir kriter ile incelemek gerekir. Bu amaçla etkinlik (effectiveness) kavramını kriter

olarak kullanabiliriz. Bir yük dengeleme algoritması, belli koşullar altında eğer sistemi yük dengelemesinin yapılmadığı durumdan daha iyi bir performansa getirebiliyorsa etkin olarak ifade edilir.

Algoritmik perspektiften bakacak olursak, eğer bir algoritma sürekli ve belirsiz bir şekilde yararsız işlemler gerçekleştiriyorsa bu algoritma dengesiz olarak ifade edilir. Örnek olarak, yükü fazla olan bir sistemde, bir transfer sonucu fazla yüklü duruma gelen bir işlemci, başka bir yük transferine sebep olacak ve bu işlem işlemciden işlemciye belirsiz bir süre devam edecek, iş sürekli bir işlemciden başka bir işlemciye aktarılacak ve hiç servis alamayacaktır.

2.3.3 Yük dağıtım algoritması parçaları

Genel olarak bir yük dağıtım algoritması dört parçadan oluşur: Hangi işlemcinin yük transferi için uygun olduğunu belirleyen *transfer politikası*, hangi işin transfer edileceğini belirleyen *seçim politikası*, yük transferinin hangi işlemciye yapılacağını belirleyen *lokasyon politikası* ve sistemin yük durumu bilgisini toplayan *bilgi politikası* (Singhal and Shivaratri, 1994).

2.3.3.1 Transfer politikası

Transfer politikaları genellikle eşik (threshold) yöntemine dayanır. Eşik değeri işlemci yük değeri cinsinden bir değerdir. Yeni bir iş geldiğinde, eğer işlemci yükü eşik değerini aşıyorsa işlemci gönderici, yani yük transferini yapacak işlemci olarak belirlenir. Eğer yük değeri

eşik deęerinin altına dūřmūřse iřlemci alıcı, yani yūk transferinin yapılacaęı iřlemci olarak belirlenir.

Bir bařka yōntem ise, yūk transferini bilgi politikası tarafından iřlemciler arası yūk durumlarında bir dengesizlik tespit edildięi anda tetikler.

2.3.3.2Seçim politikası

Seçim politikası, transfer politikasının bir iřlemciyi gönderici olarak belirlemesinin ardından transfer edilecek iři seęer. En basit yaklařım, iřlemcinin gönderici olarak belirlenmesine yol aęan, yani iřlemci yūkünün eřik deęerini ařmasına sebep olan yeni gelen iři seęmektir. Bu tūr iřlerin transfer edilmesi de kolaydır, çūnkū bu iřler henüz iřletilmeye bařlanmadıęı için bōlünmemiř(nonpreemptive)tir.

Transfer için iři seęiminde bazı kriterler bulunmaktadır. Bunlardan bir tanesi transfer iřleminin sonucunda sistemin ortalama cevap verme sūresinin dūřmesidir. Bir bařkası, iřin transfer edilmesi sonucu geręekleřen çalıřma sūresinin, iřlemin transfer edilmeden geręekleřen çalıřma sūresinden būyūk olmamasıdır. İř seęiminde, transfer iřleminin sisteme getireceęi yūkūn minimal dūzeyde tutulması, transfer edilecek iřin lokasyon baęımlı sistem çağırılarının minimal dūzeyde olması gibi bařka faktōrler de gōz ōnūne alınmalıdır.

2.3.3.3Lokasyon politikası

Lokasyon politikası yük paylaşımı için uygun işlemciler (alıcı ve gönderici) bulmakla yükümlüdür. Kullanılan yöntemlerden biri sorgulamadır (polling). Bu yöntemde bir işlemci bir başka işlemciyi yük transferi için uygun olup olmadığı konusunda sorgular. Sorgulama rastgele bir işlemci seçilerek, ya da bilgi politikasınca toplanan bilgiler doğrultusunda belirlenen bir işlemciye yapılabilir. Bir başka yöntem ise yayındır (broadcast).

2.3.3.4Bilgi politikası

Bilgi politikası, sistemin işlemcilerinden ne zaman, nereden ve hangi bilginin toplanması gerektiğinin belirlenmesinden sorumludur. Bunu için üç yöntem bulunmaktadır:

İsteğe-bağlı (demand driven): Bu yöntemde işlemci, gönderici ya da alıcı duruma geçtiği anda (transfer ve seçim politikaları sonucunda), yani yük transferi için uygun hale geçtiğinde diğer işlemcilerden bilgi toplar. Bu politika gönderici-başlatımlı, alıcı-başlatımlı, ya da simetrik-başlatımlı olabilmektedir. Gönderici-başlatımlı yöntemde gönderici yük transferi için uygun bir alıcı arar, alıcı-başlatımlı yöntemde alıcı yük transferi yapabileceği uygun bir gönderici arar, simetrik-başlatımlı yöntem diğer iki yöntemin kombinasyonudur.

Periyodik: Bu yöntemde işlemciler periyodik olarak birbirlerinden bilgi toplarlar. Toplanan bilgi doğrultusunda transfer politikası bir işlemcide yük transferine karar verir. Bu yöntem, kendini

sistem durumuna göre adapte edemez. Sistemin yük seviyesinin yüksek olduğu durumlarda işlemcilerin hemen hepsi yüklü olduğundan yük transferi yarar getirmez. Bu durumda bilgi toplamak da gereksizdir. Ancak periyodik yöntemde, bu durumda bile bilgi toplama işlemi sürdürüldüğü için sisteme gereksiz yere ek yük getirilmektedir.

Durum-değişimli(state-change-driven): Bu yöntemde, işlemciler durumları belli derecede değişime uğradığında bilgilerini diğer işlemcilere yayarlar. Bu yöntemin isteğe bağlı yöntemden farkı işlemcilerin bilgi toplamak yerine durumlarını diğer işlemcilere yaymasıdır. İşlemciler durumlarını ya merkezi bir işlemciye iletirler ya da tüm diğer işlemcilere yayarlar.

2.3.4 Yük dağıtım algoritmaları

2.3.4.1 Gönderici-başlatımlı (sender-initiated) algoritmalar

Bu tip algoritmalarda yükü eşik değerini aşan işlemci (gönderici) yükü hafif bir işlemci (alıcı) arar (Chang and Livny, 1986).

Transfer Politikası: Bu algoritmalar eşik tabanlı işlemci kuyruk uzunluğu kullanırlar. Transfer politikası yeni bir iş geldiğinde başlar. İşlemci kendini eğer yükü eşik değerinin üzerindeyse gönderici, altındaysa alıcı olarak belirler.

Seçim Politikası: Bu tür algoritmalar sadece yeni gelen işleri transfer için seçerler.

Lokasyon Politikası: Lokasyon politikasına göre üç çeşit gönderici-başlatımlı algoritma bulunmaktadır:

Rastgele: Bu yöntemde diğer işlemcilerin durum bilgileri kullanılmaz. Yük transferi rastgele seçilen bir işlemciye yapılır. Bu noktada oluşabilecek bir problem, rastgele seçilen işlemcinin de yüklü durumda bulunmasıdır. Böyle bir durumda bu işlemci de yükü başka bir işlemciye göndermeye çalışacaktır ve bu da gereksiz yük transferlerine sebep olacaktır. Bunu önlemenin bir yolu bir iş için transfer edilebilme sayısını sınırlamaktır. Bu algoritma yük dağıtımı yapmayan sistemlere göre oldukça iyi sonuçlar vermektedir.

Eşik (threshold): Rastgele yöntemde meydana gelen gereksiz yük transferi problemi, rastgele seçilen bir işlemcinin alıcı olup olmadığının sorgulanmasıyla önlenmektedir. Eğer seçilen işlemci alıcıysa yük transferi gerçekleştirilmekte, değilse başka bir işlemci seçilerek sorgulanmaktadır. Sorgulanan işlemci sayısı belli bir eşik değeriyle sınırlandırılmakta, eşik değeri aşıncaya kadar uygun bir işlemci bulunamazsa iş orijinal işlemcide çalıştırılmaktadır. Gereksiz transferleri önleyerek eşik yöntemi rastgele yönteme göre oldukça iyi sonuçlar vermektedir.

En-kısa: Bu algoritmanın diğerlerinden farkı, en iyi alıcıyı belirlemeye çalışmasıdır. Gönderici, belli sayıda işlemciyi sorgulayarak kuyruk uzunluklarını elde eder ve kuyruk uzunluğu eşik değerinin altında (alıcı durumda) olan işlemcilerden en kısa kuyruk uzunluğuna sahip olan işlemciye yük transferini gerçekleştirir. Yapılan testlerde bu algoritmanın eşik algoritmasına göre çok az bir performans artışı sağladığı görülmüştür. Bu da ayrıntılı durum bilgisi elde etmenin gereksiz olabileceğini göstermektedir.

Bilgi Politikası: Eşik ve en-kısa lokasyon politikalarında bilgi sorgulama işlemi, transfer politikası bir işlemciyi gönderici olarak belirlediğinde gerçekleştirilmektedir. Yani bilgi politikası olarak isteğe-bağlı yöntem kullanılır.

Bu tür algoritmalar sistemi, sistem yükünün fazla olduğu durumlarda dengesiz bir hale sürükleyebilirler. Sistem yüklü bir durumdayken işlemcilerin çoğu gönderici durumunda bulunduğu için bu işlemcilerin alıcı bulmak için yaptıkları sorgulamalar başarısız olacaktır. Yük arttıkça sorgulamalar da artacak ve bir noktada sistemin mevcut işlem kapasitesi büyük ölçüde sorgulama ve bu sorgulamalara cevap vermek için kullanılacaktır. Sisteme gelen yük miktarı ile yük dağıtım işleminden kaynaklanan yük miktarı sistemin kapasitesini aştığında dengesizlik oluşacaktır. Bu nedenle bu tür algoritmalar yüksek yük durumunda etkin değildirler ve sistemi dengesizliğe sürükleyebilirler.

2.3.4.2 Alıcı-başlatımlı (receiver-initiated) algoritmalar

Bu tür algoritmalarda yük dağıtım işlemi yüklü bir işlemciden yük isteyen alıcı (yük durumu hafif olan işlemci) tarafından başlatılır (Chang and Livny, 1986).

Transfer Politikası: Bu algoritmalar eşik tabanlı işlemci kuyruk uzunluğu kullanırlar. Transfer politikası, çalışmakta olan bir iş tamamlandığında tetiklenir. İşlemci eğer yükü eşik değerinin altındaysa alıcı durumuna geçer. Yükü eşik değerinin üzerinde olan işlemci ise gönderici durumundadır.

Seçim Politikası: Bu algoritma bölüm 2.3.3.2’de bahsedilen yaklaşımlardan herhangi birine göre seçim politikasını belirler.

Lokasyon Politikası: Bu algoritmada, rastgele seçilen bir işlemcinin yükünün transfer işleminin sonunda eşik değerinin altına düşüp düşmeyeceği sorgulanır. Eğer düşmeyecekse sorgulanan işlemci yük transferi yapar. Diğer durumda ise rastgele bir başka işlemci seçilerek aynı sorgulama yapılır. Bu işlem belli sayıda tekrarlanır. Eğer bu sorgulamalardan sonra da yük transferi yapılamamışsa, alıcı başka bir iş tamamlanıncaya kadar bekler ve tekrar sorgulamaya başlar. Algoritmanın başka bir şekline göre ise alıcı belli bir süre bekler ve bu süre sonunda hala alıcı durumundaysa sorgulama işlem yeniden başlatılır. Eğer ikinci yöntem kullanılmazsa, alıcının belli bir iş tamamlanıncaya kadar bekleme süresindeki boş işlem kapasitesi kaybedilmiş olacaktır.

Bilgi Politikası: İsteğe-bağlı bilgi politikası kullanılır, çünkü herhangi bir işlemci alıcı konumuna geçtiğinde bilgi toplanır.

Bu tür algoritmalar sistemi dengesizliğe sürüklemeyebilir. Çünkü sistemin yüklü olduğu durumlarda sistemde çok sayıda gönderici bulunmaktadır ve alıcıların yaptığı sorgulamalar yüksek bir başarı oranına sahip olacaktır. Bu da sorgulama mekanizmasının etkin bir şekilde kullanılmasını sağlar. Sistemin az yüklü olduğu durumlarda ise fazla sayıda olan alıcılar az sayıda bulunan göndericileri bulmaya çalışırken çok sayıda sorgulama yapmak durumunda kalacaklardır. Ancak sistemin işlem kapasitesinin büyük bölümü zaten boş olduğundan başarısız sorgulamalar sistemi dengesiz bir yapıya sürüklemeyecektir.

Bunun yanında, alıcı-başlatımlı algoritmalar, büyük ölçüde işletimine başlatılmış işlerin transfer edilmesine sebep olur. Çünkü bu

algoritmelerde alıcı göndericiden yük transferi istediği anda göndericide henüz başlatılmamış iş yoksa, çalışmakta olan bir işin transferi gerçekleştirilecektir. Bu nedenle bu algoritmalar bölünmüş yapıda (preemptive) algoritmalar ve yük transferleri zor ve karmaşıktır.

2.3.4.3 Simetrik algoritmalar

Bu tür algoritmelerde hem gönderici hem de alıcı konumundaki işlemciler transfer sürecini başlatabilirler. Bu nedenle bu algoritma önceki iki algoritmanın da avantajlarından yararlanır. Sistemin yük durumunun düşük olduğu durumlarda gönderici-başlatımlı taraf yükü hafif işlemci bulmakta başarılı olurken, yüksek sistem yükünde alıcı-başlatımlı taraf gönderici bulmakta başarılı olmaktadır. Fakat aynı zamanda bu yöntem, bu iki algoritmanın dezavantajlarına da sahiptir. Algoritma, gönderici-başlatımlı algoritmelerde olduğu gibi yüksek yük durumunda sistemi dengesizliğe sürükleyebilirken, alıcı-başlatımlı algoritmelerdeki işin yarıda kesilip transfer edilmesinden kaynaklanan transfer işlemi zorluğuna da sahiptir (Singhal and Shivaratri, 1994).

Bu tür algoritmalara örnek olarak, above-average algoritmasını (Krueger and Finkel, 1984) verebiliriz. Bu algoritma, her bir işlemciye yük seviyesini belli bir kabul edilebilir aralıkta tutmaya çalışır.

Transfer Politikası: Transfer politikası eşik yöntemini kullanır. Bunun için iki adet adapte edilebilir eşik değeri bulunmaktadır. Bu eşikler bir işlemcinin sistemin yük durumu hakkında yaptığı tahmine göre belirlenir. Örnek olarak, eğer işlemci tüm sistemin yükünü 2 olarak tahmin ettiyse küçük eşik 1 büyük eşik ise 3 olacaktır. Yükü küçük eşikten düşük olan işlemci alıcı, büyük eşikten büyük olan işlemci ise

gönderici olarak belirlenir. Yük durumu iki eşik arasında olan işlemciler alıcı ya da gönderici olarak belirlenmezler.

Seçim Politikası: Bu algoritma bölüm 2.3.3.2’de bahsedilen yaklaşımlardan herhangi birine göre seçim politikasını belirler.

Lokasyon Politikası: Lokasyon politikası iki parçadan oluşur:

Gönderici-başlatımlı parça:

- Gönderici, *TooHigh* mesajı yayınlar, *TooHigh* zaman-aşımı (timeout) alarmını başlatır ve zaman aşımı oluşuncaya kadar *Accept* mesajı bekler.
- *TooHigh* mesajı alan alıcı *TooLow* zaman-aşımını iptal eder, *TooHigh* mesajını gönderen işlemciye *Accept* mesajı gönderir, yük değerini artırır ve *AwaitingTask* zaman-aşımı alarmını başlatır. Eğer transfer işlemi gerçekleştirmeden *AwaitingTask* zaman-aşımı gerçekleşirse yük değerini tekrar eski haline alır.
- *Accept* mesajı alan işlemci eğer hala gönderici durumunda ise transfer işlemi için en uygun işi seçer ve mesajı gönderen işlemciye transfer eder.
- *TooHigh* zaman-aşımı dolduğunda gönderici eğer hiç *Accept* mesajı almadıysa, sistemin yük değeri tahmininin çok düşük olduğunu anlar ve bu durumu düzeltmek, yani diğer işlemcilerin yük tahminlerini yükseltmeleri için *ChangeAverage* mesajı yayınlar.

Alıcı-başlatımlı parça:

- Alıcı *TooLow* mesajı yayınlar, *TooLow* zaman-aşımı alarını başlatır ve *TooHigh* mesajı bekler.
- Eğer alıcı *TooHigh* mesajı alırsa, gönderici-başlatımlı parçada bahsedilen uzlaşma sürecini gerçekleştirir.
- Eğer *TooHigh* mesajı almadan *TooLow* zaman-aşımı oluşursa, alıcı diğer işlemcilerdeki ortalama yük tahmini değerini düşürmek için *ChangeAverage* mesajı yayınlar.

Bilgi Politikası: İsteğe-bağlı bilgi toplama yöntemini kullanır. Burada altı çizilecek bir nokta, ortalama sistem yük değerinin her bir işlemcide belirlenmesiyle, küçük bir miktar performans kaybı yaratılıp, mesaj karmaşıklığının azaltılmasının sağlanmasıdır. Bir başka nokta ise, kabul edilebilir aralık sayesinde sistemin yük dağıtım işleminin iletişim ağının yoğunluğuna göre adapte olabilmesidir. İletişim ağının yüklü/yüksüz durumuna göre (mesaj iletim sürelerine bakılarak belirlenir) kabul edilebilir aralık büyütülüp/küçültülerek algoritmanın kendini adapte etmesi sağlanır.

2.3.4.4Adaptif algoritmalar

Bu tür algoritmalarla örnek olarak dengeli simetrik bir algoritma örnek verilebilir (Singhal and Shivaratri, 1994). Diğer algoritmalarındaki, yük paylaşımından kaynaklanan dengesizliğin sebebi göndericinin yaptığı başarısız sorgulamalardır. Bu algoritma, sorgulamalardan elde edilen bilgilerden yararlanarak (diğer algoritmalar bu bilgileri kullanmazlar) işlemcileri alıcı, gönderici ve normal olarak sınıflandırır. Bu üç çeşit sınıflandırma bilgileri her bir işlemcideki üç ayrı listede

tutulur (gönderici listesi, alıcı listesi ve normal işlemci listesi). Bu listeler, üzerinde güncelleme ve arama işlemleri çok az işlemci gücü gerektirecek şekilde özel olarak tasarlanmıştır. Başlangıçta tüm işlemciler alıcı olarak kabul edilirler ve alıcı listesine yerleştirilirler, diğer listeler boştur.

Transfer Politikası: Transfer politikası, işlemci kuyruk uzunluğunu kullanan eşik yöntemidir. Transfer politikası yeni bir iş geldiğinde ya da işletilen bir iş tamamlandığında aktive olur. Transfer politikası iki ayrı eşik değeri kullanır. Eğer bir işlemcinin yük değeri küçük eşikten düşükse alıcı, büyük eşikten yüksekse gönderici, iki değer arasındaysa normal olarak belirlenir.

Seçim Politikası: Bu algoritmanın gönderici-başlatımlı parçası sadece yeni gelen işleri seçer. Alıcı-başlatımlı parçası ise bölüm 2.3.3.2'de bahsedilen yaklaşımlardan herhangi birine göre seçim politikasını belirler.

Lokasyon Politikası:

Gönderici-başlatımlı parça:

Gönderici başlatımlı parça, bir işlemci gönderici durumuna geçince tetiklenir. Gönderici, alıcı listesinin en başındaki işlemciyi sorgular. Sorgulanan işlemci, sorgulayan işlemciyi bulunduğu listeden çıkararak gönderici listesinin başına ekler ve o anki yük durumuna göre kendi durumunu göndericiye bildirir. Gönderici, eğer sorguladığı işlemci alıcı ise yeni gelen işi bu işlemciye transfer eder. Eğer sorgulanan işlemci alıcı değilse, gönderici bu işlemciyi alıcı listesinden çıkarır ve bildirdiği yük durumuna göre diğer iki listeden birinin başına ekler. Ardından alıcı

listesinin en başındaki işlemciyi sorgular. Sorgulama süreci, gönderici bir alıcı buluncaya, belli bir sorgulama sayısı aşıncaya ya da alıcı listesi boşalincaya kadar devam eder. Eğer sorgulama süreci başarısızlıkla sonuçlanırsa iş göndericide başlatılır.

Alıcı-başlatımlı parça:

Alıcı-başlatımlı parçanın amacı yükünü alabileceği uygun bir gönderici bulmaktır. İşlemciler şu sırada sorgulanır: Gönderici listesi baştan sona doğru (en güncel bilgiyi kullanarak), normal işlemci listesi sondan başa doğru (en eski bilgiyi kullanarak), alıcı listesi sondan başa doğru.

Alıcı-başlatımlı parça bir işlemci alıcı konuma geçtiğinde tetiklenir. Alıcı, yukarıda belirtilen şekilde seçilen işlemciyi sorgular. Sorgulanan işlemci eğer göndericiyse üzerindeki bir işi alıcıya transfer eder ve transfer sonrası yük durumunu alıcıya bildirir. Eğer sorgulanan işlemci gönderici değilse, sorgulayan işlemciyi bulunduğu listeden çıkarır ve alıcı listesinin başına ekler ve alıcıya o anki yük durumunu bildirir. Alıcı sorgulanan işlemciden cevap alınca, bu işlemciyi bulunduğu listeden çıkarır ve uygun listenin başına ekler.

Sorgulama işlemi bir gönderici bulunduğu anda, alıcının durumu değiştiğinde (gönderici ya da normal komuna geçtiğinde) veya belli sayıda sorgulama yapıldığında sona erer.

Bilgi Politikası: Bilgi politikası, bir işlemci gönderici ya da alıcı konuma geçtiği anda başladığı için isteğe-bağlı bilgi toplama yöntemidir.

Sistem yükünün fazla olduğu durumlarda, gönderici-başlatımlı kısım başarısız sorgulamalarda bulunacaklardır. Bu sorgulamalar sonucunda, sorgulanan işlemciler alıcı listesinden çıkarılacaklardır ve en sonunda alıcı listesi boşalacaktır. Alıcı-başlatımlı kısım başarısız sorgulamalarda bulunmadıkça alıcı listesi boş kalacaktır. Alıcı listesi boş kaldıkça gönderici-başlatımlı kısmın çalışması durmuş olacaktır, çünkü gönderici yalnızca alıcı listesindeki işlemcileri sorgulamaktadır. Sonuç olarak yüksek sistem yükünde göndericilerin gereksiz sorgulamalar yaparak sistemi dengesizliğe sürüklemesi önlenmiş olacaktır. Bu durumlarda sadece alıcı-başlatımlı parça çalışır durumda olacaktır.

Sistem yükünün az olduğu durumlarda alıcı-başlatımlı kısım başarısız sorgulamalarda bulursa da, bu sorgulamalar sistem yükü düşük olduğundan performans kaybına neden olmayacak, bunun yanısıra alıcı listelerinin güncel tutulmasını sağlayacaktır. Alıcı listelerinin güncel olması da gönderici-başlatımlı sorgulamaların başarı oranını yükseltecektir.

Sonuç olarak yüksek yük durumunda alıcı-başlatımlı, düşük yük durumunda gönderici-başlatımlı, ortalama yük durumlarında da simetrik politikalar kullanılarak sistemin geniş bir yük seviyesi aralığında performansı optimum düzeyde tutulacak ve dengeli yapısı korunacaktır.

2.3.5 Performans karşılaştırması

Şekil 2.1'de de görüldüğü gibi rastgele lokasyon politikası kullanan gönderici-başlatımlı algoritmalar (şekilde RANDOM olarak belirtilmiştir), hiç yük dağıtımı yapmayan bir sistemle (şekilde M/M/1 olarak belirtilmiştir) karşılaştırıldığında uyguladığı bu basit yöntemle bile