

Gezgin Etmen Sistemlerinin Başarım Ölçümü: Benzetim Tekniği

Gürol Erdoğan¹, Mustafa Yıldız¹, Mehmet Erdem Türsem², Selahattin Kuru¹

¹ Enformatik Uygulama ve Araştırma Merkezi, Işık Üniversitesi, İstanbul

² Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, İstanbul

Özet. Gezgin yazılım etmenleri, özellikle dağınık uygulamalar ve heterojen sistemleri kapsayan Internet uygulamalarında sıklıkla kullanılan yeni bir yaklaşımdır. Gezgin yazılım etmenlerinin başarımları, bu tür sistemlerin fizibilitesi için önemli bir ölçüttür. Bu çalışmada gezgin yazılım etmeni uygulamalarının büyük bölümünü kapsayan üç farklı jenerik etmen modelinin her biri için etmenlerin başarımlarını değerlendirecek kesikli olay tabanlı bir benzetim modeli geliştirilmiştir. Bu modeller tek adım, çok adım ve buluşma etmen modelleridir. Bu benzetim modelleri daha sonra bir benzetim yazılımı üzerinde uygulanmış, her bir etmen modeli için toplam servis süreleri elde edilmiştir. Benzetimde kullanılan parametre değerleri literatürde varolan bilgiler gözetilerek belirlenmiştir.

1 Giriş

Gezgin yazılım etmenleri, amaçlarını gerçekleştirmek için, kendi kodunu ve verisini bir bilgisayardan başka bir bilgisayara aktarma özelliğine sahip yazılımlardır. Gezgin yazılım etmen sistemleri, özellikle dağınık uygulamalar ve heterojen platformlarda yoğun olarak kullanılmaktadır. Birçok uygulama alanında, işlem yükünü verinin bulunduğu uzaktaki bilgisayara aktararak iletişim maliyetlerini düşürmek için kullanılan bu tür sistemlerde başarımlar son derece önemlidir. Sistem tasarımı ve fizibilite çalışmalarında, etmen sistemlerinin başarımları göz önünde tutulmak zorundadır. İstemci / sunucu mimarisinin sorunlarına çözüm getiren etmen sistemleri, istemci / sunucu mimarisinin aksine, daha ölçeklenebilir, daha sağlam, kolay uyarlanabilen ve özelleştirilebilen bir yapıya sahiptir.

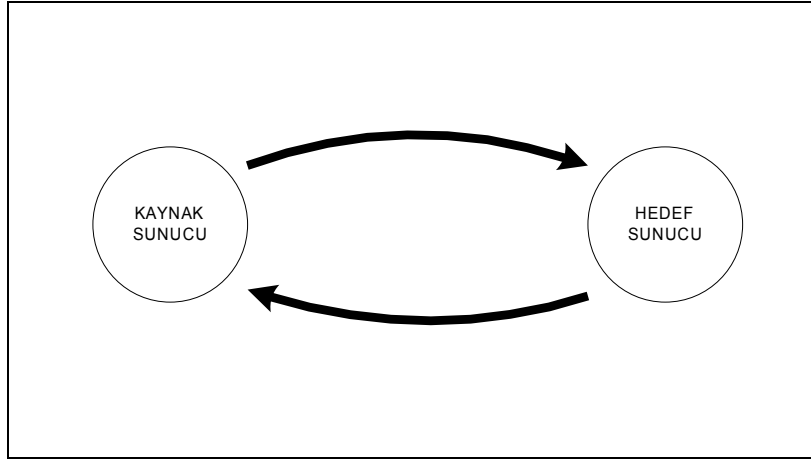
Gezgin yazılım etmenlerini kullanmanın sağladığı yararlar iki temel grupta toplanabilir; yazılım mühendisliğine ilişkin yararlar ve başarıma ilişkin yararlar. Başarıma ilişkin yararlar, daha etkin bant genişliği kullanımı, daha az işlem yükü, daha az gecikme ve daha iyi hata toleransı olarak sıralanabilir. Yazılım mühendisliğine ilişkin yararlar ise, uygulama geliştiricilerine, bilinen uygulama çeşitleri için daha doğal tasarım çözümleri sunarak esneklik sağlaması, kodun yeniden kullanılabilirliğini arttırması ve daha modüler kod geliştirilmesine olanak sağlaması ve heterojen platformların aynı sistem içerisinde kullanılabilmesini sağlaması olarak özetlenebilir.

Yazılım etmeni sistemlerinin başarımların ölçümleri üç farklı şekilde yapılabilir; analitik yöntem, benzetim yöntemi ve gerçek zamanlı ölçümler. Bu yöntemler arasında kolay ve kısa zamanda uygulanabilirlik ve maliyet unsurları göz önünde bulundurulduğunda benzetim yöntemi çok uygun bir yöntem olarak göze çarpmaktadır. Bu çalışmada, yazılım etmeni uygulamalarının büyük bölümünü kapsayan üç farklı jenerik etmen modelinin her biri için, gezgin etmenlerin başarımlarını değerlendirecek kesikli olay tabanlı bir benzetim modeli geliştirilmiştir. Bu modeller tek adım, çok adım ve buluşma etmen modelleridir. Bu benzetim modelleri daha sonra bir benzetim yazılımı üzerinde uygulanmış, her bir etmen modeli için toplam servis süreleri elde edilmiştir. Benzetimde kullanılan parametre değerleri literatürde varolan bilgiler gözetilerek belirlenmiştir.

2 Jenerik Etmen Modelleri

Gezgin yazılım etmenlerinin başarımların ölçümü, etmen sistemlerinin ve sistem parametrelerinin oldukça çeşitli olmasından dolayı zordur. Bu parametreler; etmen yazılımının uygulanma teknikleri, çalışma ortamları, ağ yapısı, ağ yükü, tasarım prensipleri ve yazılımın geliştirildiği programlama diline has parametreler olarak özetlenebilir. Etmen sistemlerinin başarımların analizi; analitik teknikler, fiziksel ölçüm ve benzetim yöntemleri kullanılarak yapılmaktadır. [1,2,3,4,5] Bu çalışmada, gezgin yazılım etmenlerinin başarımların ölçümlerinde “kesikli olay tabanlı benzetim çerçevesi” ortaya konulmuştur. Analiz edilen jenerik etmen modelleri, etmenlerin hareket etme şekillerine göre sınıflandırılan tek adım, çok adım ve buluşma modelleridir.

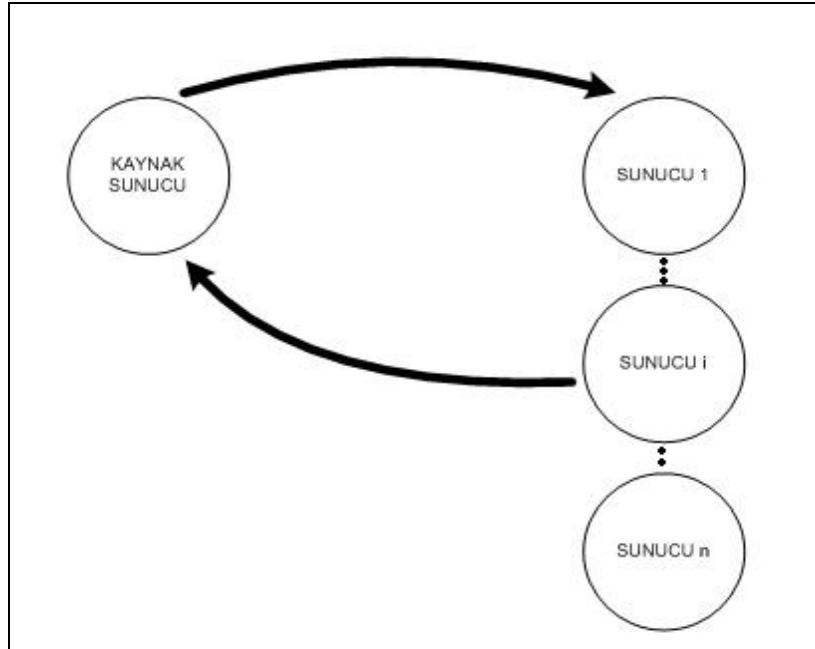
Tek adım etmen modeli[6], A etmeninin, birbirine heterojen bir iletişim kanalı ile (LAN veya Internet) bağlı S_H sunucusundan, S_D sunucusuna gidip geri dönmesi şeklinde bir yapıdadır. Gidiş-dönüşten kasıt, yazılımın S_D sunucusuna taşınması, asıl çalışmasını gerçekleştirip, S_H sunucusuna geri dönmesidir. Modelin yapısı Şekil 1’de gösterilmiştir.



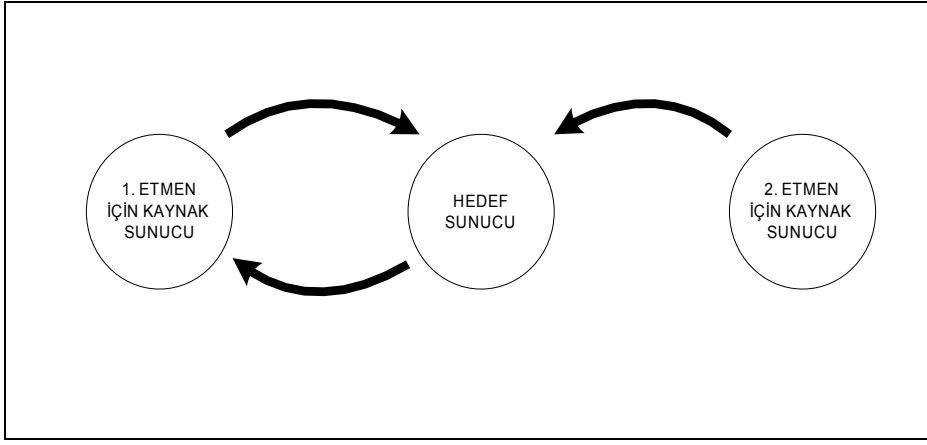
Şekil 1. Tek adım etmen modeli

Çok adım etmen modeli [6], N adet sunucu içerisinde, bir tanesini arayıp bulan ve o sunucu üzerinde birtakım işlemler gerçekleştirip kendi sunucusuna dönen etmen modelidir. Burada önemli olan etmenin aradığı sunucuya has özelliklerini bilmesi ve N sunucu içinden hedeflediği sunucuyu ayırt edebilmesidir. Şekil 2 bu modelin yapısını göstermektedir.

Son etmen modeli, gezgin etmenlerin birbirini bulması şeklinde gerçekleşen buluşma modelidir.[6] Bu modelde etmen, S_H sunucusundan yollar ve arada bir sunucuya (S_M) ulaşır. Bu arada S_H sunucusundan, servis sunucusuna (S_S) bir servis etmeni istemi yollar. S_S sunucusu, S_M sunucusuna A_S etmenini gönderir. Her iki etmen de S_M sunucusuna ulaştığı anda çalışma başlar. Etmenlerden birinin S_M sunucusuna erken ulaşması halinde, bu etmen, diğer etmeni beklemek durumundadır. Şekil 3'te bu model gösterilmiştir. Bu çalışmada kullanılan benzetim yönteminde, veri aktarımının süresi için iki farklı değer kullanılmıştır. Bu modelin bir başka çeşidi de S_H sunucusundan yola çıkan etmenin, S_M sunucusunda birden fazla etmenle buluşması olabilir. Bu model, etmenlerin ortak bir sunucuda buluşup, güvenli iletişim kurmalarına olanak vermesi bakımından oldukça yararlıdır.



Şekil 2. Çok adım etmen modeli



Şekil 3. Buluşma etmen modeli

3 Jenerik Etmen Modellerinin Kesikli Olay Tabanlı Benzetimi

Bir gezgin yazılım etmeni üç bölümden oluşur. Bunlar, etmenin asıl algoritmasının bulunduğu kod bölümü, çalışma yığını ve etmenin evrensel değişkenlerinin değerlerini oluşturan veri bölümüdür. Etmenin sahip olduğu veri, etmen sunucularda gezdikçe ve veri topladıkça değişir ve gelişir. Etmen hareket ettiğinde bu üç bölüm birlikte hareket ederler.

Etmen hareket etme komutunu aldığı anda, etmenin kodunu ve verisini içeren bir mesaj oluşturulur. Etmen verisi, yine etmenin kendi kendine kontrol edebildiği Java nesnelerinden oluşur. Java’da bulunan serileştirme mekanizması sayesinde bu nesneler byte dizisine dönüştürülüp mesaja eklenebilmektedir. Daha sonra bu mesaj TCP/IP üzerinden hedefe gönderilir. Java’nın sınıf yükleme (ClassLoader) özelliği sayesinde mesajın içindeki serileştirilmiş nesneler hedefte dinamik olarak yeniden oluşturulur. Etmenin bu tür bir tek yönlü hareketinde başarıyı etkileyen kritik süreçler aşağıda sıralanmıştır:

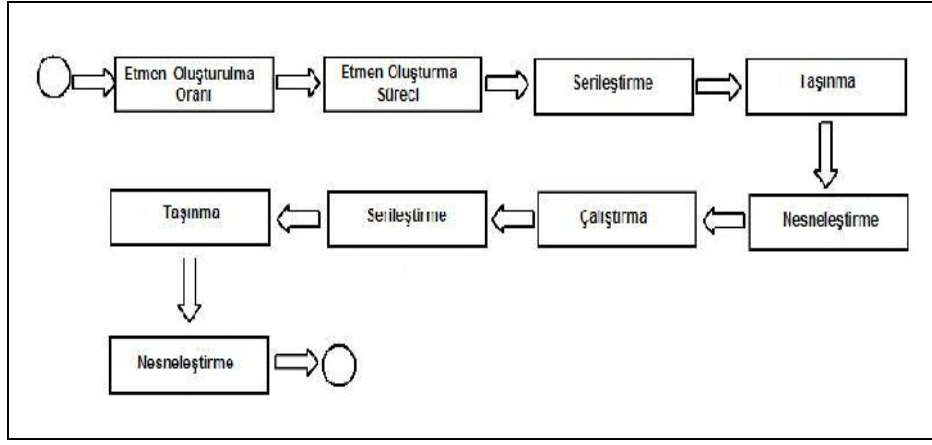
- Etmenin oluşturulması
- Etmenin serileştirilmesi
- Mesajın hedef sunucuya gönderilmesi (ağ transferi)
- Etmenin hedef sunucuda yeniden nesneleştirilmesi
- Gelen etmenin kodunun yüklenmesi ve çalıştırılması

Etmenlerin oluşturulma, serileştirilme / nesneleştirilme ve taşınma sürelerini hesaplarken uygun olasılık dağılımı fonksiyonları kullanılmıştır. Söz konusu süreçlerde kullanılan olasılık dağılımı fonksiyonları Tablo 1’de sıralanmıştır.

Tablo 1. Süreçlerde kullanılan olasılık dağılımı fonksiyonları

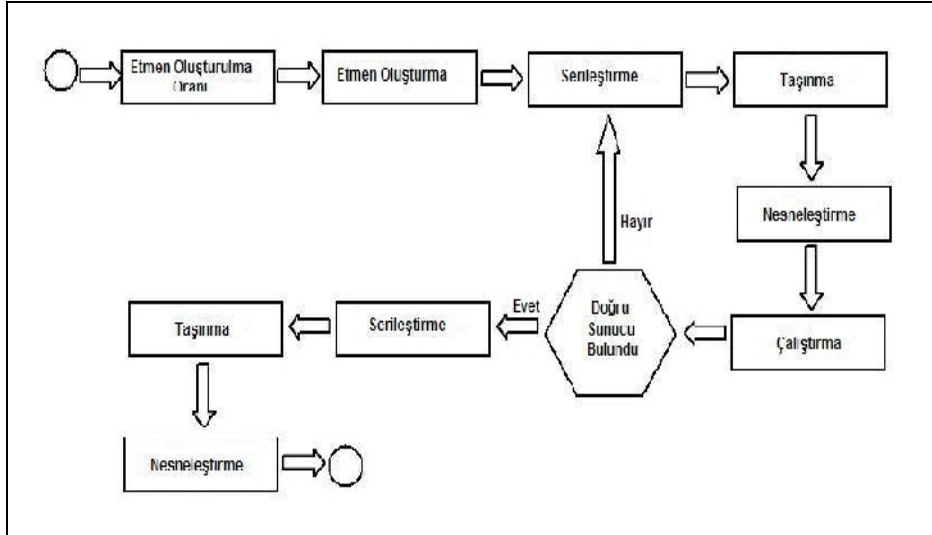
Süreç	Olasılık Dağılımı Fonksiyonu
Etmen Oluşturulma Oranı	Poisson Dağılımı
Etmen Oluşturulması	Normal Dağılım
Etmen Serileştirme	Normal Dağılım
Etmen Taşınması (LAN)	Normal Dağılım
Etmen Taşınması (Internet)	Weibull Dağılımı
Çalışma	Normal Dağılım
Etmen Nesneleştirilmesi	Normal Dağılım

Tek adım benzetim modelinde kullanılan süreçlerin üçü, etmen sunucuların üzerinde iken oluşan durumu ifade etmektedir. Bu durumlar, etmenin oluşturulması, etmenin ulaşması ve etmenin serileştirilmesidir. Bu süreçleri, etmenin taşınmasını ifade eden bir süreç ve etmenin çalıştırılmasını ifade eden başka bir süreç takip eder. Çalışma bittikten sonra etmenin geri dönmesi, taşınmayı ifade eden daha önceki sürecin aynısıdır. Etmen geri döndüğünde son olarak nesneleştirme sürecinden geçer. Bu modelde oluşan süreçlerin sıralaması Şekil 4’te gösterilmiştir.



Şekil 4. Tek adım benzetim modeli

Çok adım benzetim modeli, etmenin oluşturulması ve taşınması süreçleri bakımından aynıdır. İlk modelden farklı olarak, bu modelde etmen, doğru sunucuda olup olmadığını anlamak için bir denetleme yapar. Eğer etmen istenilen hedef sunucuda değil ise, serileştirme işlemi tekrar yapılır ve etmen başka bir sunucuya taşınır. Doğru hedef sunucu bulunduğu etmen tekrar serileştirilir ve kaynak sunucuya geri döner. Benzetim modelinde etmenin doğru sunucuyu kaç dolaşımında bulunduğu, özbiçimli dağılım olarak belirtilmiştir. Şekil 5'te bu modele ait süreçlerin sıralaması verilmiştir.



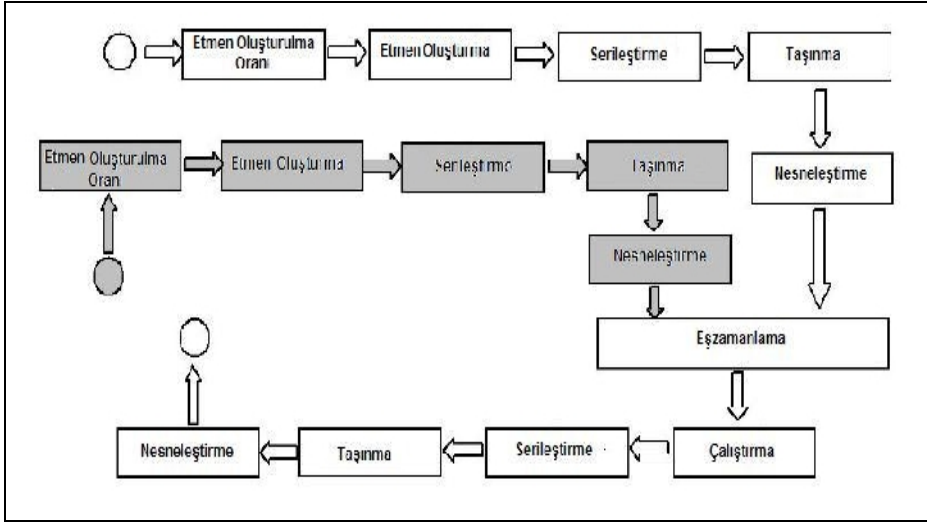
Şekil 5. Çok adım benzetim modeli

Buluşma modeli de gerçekleşen süreçler bakımından tek adım modeline benzer ve buna ek olarak hedef sunucuda, ilk etmenle ikinci etmen arasında geçen bir eşzamanlama süreci geçirir. Çalışma bittikten sonra ilk etmen kendi sunucusuna geri taşınır ve süreçler tamamlanır.

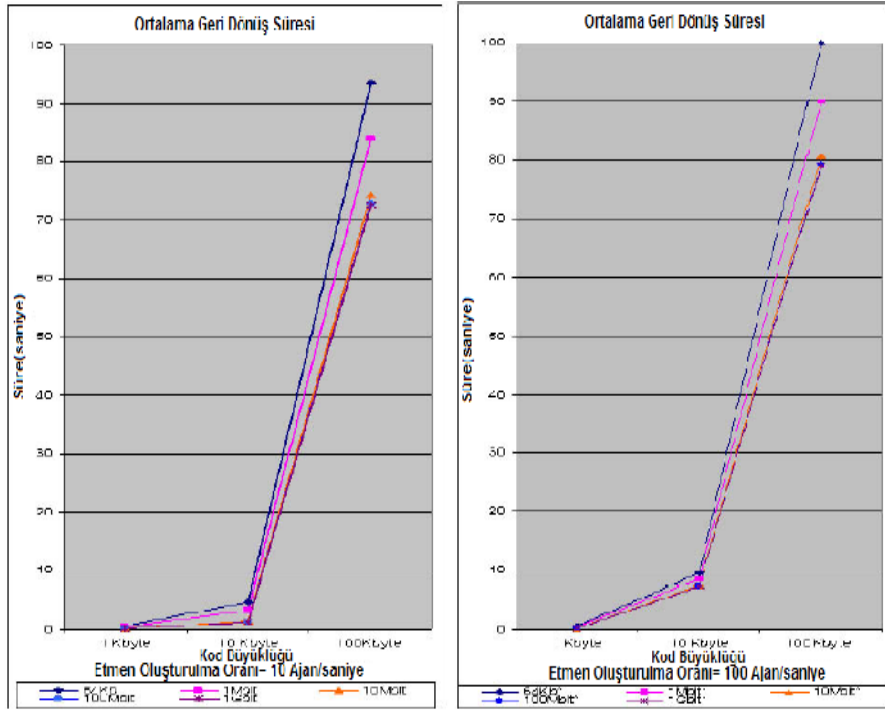
4 Benzetim Tasarımı ve Sonuçlar

Bu çalışmada yapılan benzetimlerde etmenlerin oluşturulma oranları 10 etmen/saniye ve 100 etmen/saniye olarak iki farklı parametre kullanılmıştır. Etmenlerin kod büyüklüklerinin 1Kbyte, 10Kbyte ve 100Kbyte olarak değiştiği varsayılmıştır. İletişim kanallarının bant genişliği etmenlerin taşınma sürelerini etkileyen önemli bir faktördür. Benzetimlerde kullanılan bant genişlikleri, 64KByte, 1Mbit, 10Mbit, 100Mbit ve 1Gbit'tir. Arama modelinde kullanılan istemci sayısı 5, 10, 50, 100, 500 olarak değişmektedir. Sonuç olarak, her bir modelde etmenlerin çalışıp ortalama geri dönme süreleri sunulmuştur.[6]

Şekil 7'de tek adım modelinin 10 etmen/saniye ve 100 etmen/saniye durumları için sonuçlar verilmiştir. Değişik kod büyüklükleri için zaman değişimleri incelendiğinde 10Kbyte değerinin kritik nokta olduğu gözlemlenmiştir. Tek adım durumu için etmen oluşturulma oranlarının geri dönme sürelerine çok fazla etki etmediği gözlemlenmiştir.



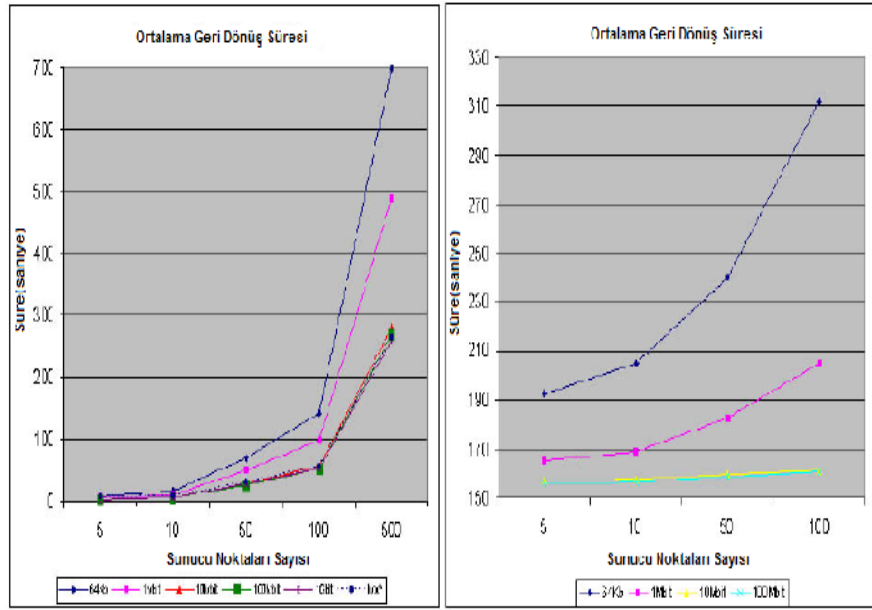
Şekil 6. Buluşma benzetim modeli



Şekil 7. Tek adım modelinde etmenin geri dönme süreleri

Çok adım modelinde etmenlerin geri dönüş süreleri, her iki etmen oluşturulma oranı için, sonucu noktalarının sayıları ile birlikte Şekil 8'de verilmiştir. Bu durumda etmen oluşturulma sürelerinin bir şekilde geri dönme sürelerini etkilediği gözlemlenmiştir.

Buluşma modeli için sonuçlar Şekil 9'da verilmiştir. Bu modelde de etmen oluşturulma sürelerinin geri dönme sürelerini etkilediği gözlemlenmiştir.

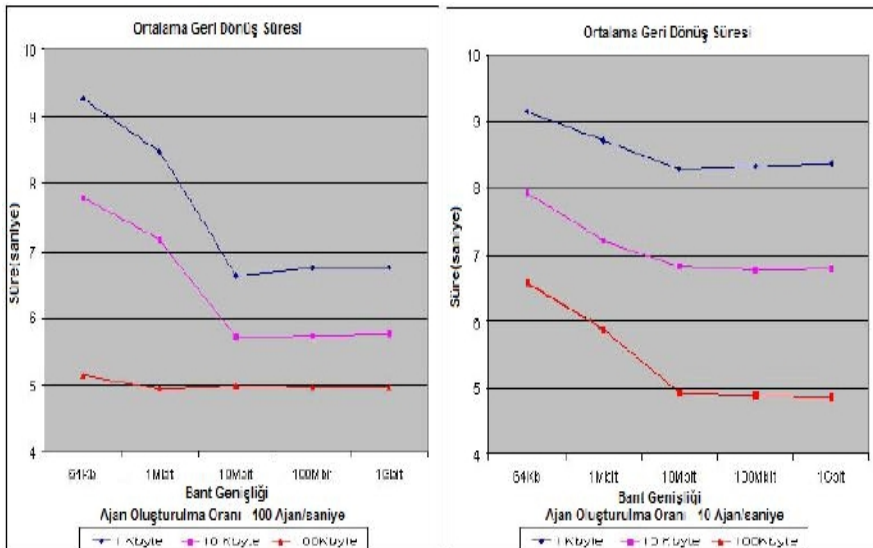


Şekil 8. Çok adım modelinde etmenin geri dönme süreleri

5 Sonuç

Yazılım etmenlerinin başarımlarını ölçümü için üç farklı metot kullanılmaktadır. Bunlar, analitik ölçümler, gerçek sistem ölçümleri ve kesikli olay ölçümleridir. Kesikli olay ölçümleri özellikle yazılım etmenleri için esnek, yeniden üretilebilir ve ucuz bir başarımlar ölçümü yöntemidir.

Bu çalışma, üç farklı etmen taşınma yöntemi için üç kesikli olay tabanlı benzetim yöntemi öne sürmektedir. Bu modellerin her biri için benzetim şemaları oluşturulmuş ve bu şemalar birçok parametre kullanılarak bir benzetim yazılımı üzerinde gerçekleştirilmiştir. Benzetim modelleri ve dağılım parametreleri literatürde varolan veri kullanılarak oluşturulmuştur. Varolan jenerik etmen modelleri için oluşturulan benzetim modelleri, diğer etmen modellerinin incelenmesi için uygun bir ortam sağlamıştır.



Şekil 9. Buluşma modelinde etmenin geri dönme süreleri

Kaynakça

1. S. Bandyopadhyay and K. Paul, "Evaluating the Performance of Mobile Agent-Based Message Communication among Mobile Hosts in Large Ad Hoc Wireless Network," Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, sayfa 69-73. ACM, Ağustos 1999

2. D. Kotz, G.Jiang, R.Gray, G. Cybenko, and R.A. Peterson, "Performance Analysis of Mobile Agents for Filtering Data Streams on Wireless Networks," Proceedings of the Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, sayfa 85-94. ACM Press, Ağustos 2000
3. L.Ismail and D.Hagimond, "A Performance Evaluation of the Mobile Agent Paradigm. OOPSLA'99," Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications, sayfa 306-313. ACM, Kasım 1999
4. A. Puliafito, S.Riccobene, and M.Scarpa, "An Analytical Comparison of the Client-Server, Remote Evaluation and Mobile Agent Paradigms," Proceedings of the Joint Symposium ASA/MA '99. First International Symposium on Agent Systems and Applications (ASA' 99). Third International Symposium on Mobile Agents (MA' 99), sayfa 278-292. IEEE-Computer Society, Ekim 1999
5. L.M.Silva, G.Souares, P.Martins, V.Batista, and L.Santos, "Comparing the Performance of Mobile Agent Systems: a Study of Benchmarking," Computer Communications, 23(8):769-778, 2000
6. M.E.Türsem, Performance Evaluation of Mobile software Agents, Yüksek Lisans Bitirme Tezi, Boğaziçi Üniversitesi, İstanbul, 2002