Veri Sıkıştırma Algoritmalarının Karşılaştırılması: Katılım Bankası Örneği

Mustafa Enliçay[1], Özgür Şahin[1], İlker Ülger[1], Ökkeş Emin Balçiçek[1], Mehmet Vacit Baydarman[1] ve Şakir Taşdemir[2]

[1] Kuveyt Türk Katılım Bankası, Ar-Ge Merkezi, Kocaeli, Türkiye
{mustafa_enlicay, ilker.ulger, emin.balcicek, ozgur.sahin, mehmet.baydarman}@kuveytturk.com.tr

[2] Selçuk Üniversitesi Teknik Bilimler Meslek Yüksekokulu, Konya, Türkiye
stasdemir@selcuk.edu.tr

Özet

Bilgi iletişim teknolojilerindeki hızlı gelişmelere paralel olarak, her gün yeni teknolojiler ve algoritmalar ortaya çıkmaktadır. Bundan dolayı bu alandaki gelişmelerin takip edilmesi, denenmesi ve daha iyi olanların sisteme dâhil edilmesi gerekmektedir. Bu çalışmada, yaklaşık 300 şubesi bulunan bir katılım bankası sisteminin ağ trafiğinde iki farklı sıkıştırma algoritması karşılaştırılmıştır. Bu algoritmalar Deflate algoritması ve bu banka sisteminde kullanılmakta olan Gzip algoritmasıdır. Karşılaştırma parametreleri olarak, farklı veri miktarlarında, sıkıştırma performansı, hız ve süre dikkate alınmıştır. Bu iki algoritma kullanılarak farklı boyutlardaki dokümanlar üzerinde testler yapılmıştır. Ayrıca bu sistemin ağ trafiğindeki veriler kullanılarak bu sisteme özel bir karşılaştırma da yapılmıştır. Bu testler sonucunda Deflate algoritmasının Gzip algoritmasına yakın sonuçlar verdiği görülmüş ve var olan algoritmanın değiştirilmesine ihtiyaç duyulmamıştır.

Anahtar Kelimeler: Deflate, Gzip, Katılım Bankası, Veri Sıkıştırma Algoritması, Yazılım

1. Giriş

Disk kapasitesi ve bellek sorunlarının eskisi kadar yaşanmadığı günümüzde, verilerin belleklerde sıkıştırılarak saklanması ihtiyacı da azalmıştır. Bunun yanında bellek miktarlarının artması, dosya ve veri boyutunun da bununla orantılı olarak artmasına neden olmuştur. Ağ üzerinden dosya transferinin yaygınlaşması, ağ üzerinde gönderilen verilerin sıkıştırılmasını çok önemli bir konuma getirmektedir. Özellikle birçok yerde şubesi bulunan büyük sistemlerde ise şubeler arasında ve merkez ile şubeler arasında veri transferi yapılırken verilerin hızlı taşınması ve band genişliğinin kontrol altında tutulması için verilerin sıkıştırılarak taşınması çok daha önemli bir konuma gelmektedir. Tüm bu sebeplerden dolayı veri sıkıştırma algoritmaları, büyük bilgisayar sistemlerinde aktif olarak kullanılmaktadır.

Veri sıkıştırma yöntemleri, sıkıştırma biçimlerine göre kayıplı ve kayıpsız sıkıştırma olarak iki grup altında incelenebilir. Kayıplı sıkıştırma yöntemlerinde, verilerin sıkıştırıldıktan sonra, orijinal verinin kısmen kaybedildiği ve tekrar açılmak istendiğinde eski verinin tam olarak elde edilemediği yöntemlerdir. Kayıpsız sıkıştırma yöntemlerinde ise verinin orijinal haline ulaşmak mümkündür.

İki tip kayıpsız sıkıştırma yöntemi vardır. Bunlar olasılık tabanlı kodlama ve sözlük tabanlı kodlamadır. Olasılık tabanlı kodlamada, sıkıştırılan verinin bütünü içinde daha sık kullanılan sembollere daha küçük boyutta bitlerden oluşan kodlar atanması prensibi ile sıkıştırma yapılır. En çok kullanılan olasılık tabanlı teknikler; Huffman kodlaması ve Aritmetik kodlamadır. Sözlük tabanlı kodlamada ise, sık tekrarlanan sembol grupları için tek bir sembol kullanılması ile sıkıştırma yapılır. En çok kullanılan sözlük tabanlı teknikler, LZ77, LZ78 ve LZW yöntemleridir. Hem olasılık tabanlı kodlamayı, hem de sözlük tabanlı kodlamayı bir arada kullanan algoritmalar, daha yüksek sıkıştırma oranları sağlarlar. DEFLATE algoritması ise Huffman ve LZ77 sıkıştırma tekniklerini bir arada kullanan bir algoritmadır [1, 2, 3].

Veri işleme teknolojilerinin geniş yelpazesinde, veri sıkıştırma ve açma işlemleri çok önemli bir teknoloji haline gelmiş bulunmaktadır. Bu teknolojiler sayesinde geçerli disk kapasitelerinde ve girdi-çıktı işlemlerinin bant genişliğinde belirgin gelişmeler sağlanmaktadır. Bu sayede veri merkezi masrafları azalmakta ve uygulamalarda hız kazandırılmaktadır [4].

Gzip algoritması bilgi teknolojilerinin farklı alanlarında yaygın olarak kullanılmaktadır. Bunlardan bir tanesi GZIP' in Dijital televizyon uygulamalarında çözücü olarak kullanılmasıdır[5].

Diğer bir kullanım alanı ise konum bazlı servislerdir. Bu servislerin gelişimiyle birlikte konumsal verinin kablosuz ağlarla taşınması bu alanda yapılan araştırmaların odak noktası olmuştur. Bu alandaki kilit teknolojilerden olan veri sıkıştırma teknolojileri araştırıldığında gelişmiş sıkıştırma metodu olan Gzip'in yaygın kullanımı göze çarpmaktadır [6].

Bu çalışmada, yaklaşık 300 şubesi bulunan bir katılım bankasında, şubeler arasındaki iletişimi hızlandırmak için farklı sıkıştırma algoritmaları kullanılarak bir yazılım uygulaması gerçekleştirilmiştir.

2. Materyal ve Metot

Veri sıkıştırmada yazılım ve donanım kullanmak suretiyle iki farklı yöntem kullanılmaktadır. Bu yöntemler kullanarak veri sıkıştırma işlemini gerçekleştiren farklı sektörler bu sayede hem zaman hem de hız olarak kazanç sağlamaktadırlar. Ekonomide önemli bir yer tutan bankacılık ve finans sektöründe de bu durum incelendiği zaman veri sıkıştırma işlemlerinin çok yaygın olarak kullanıldığı görülmüştür.

Bu çalışmada veri ölçümü yapılırken, Gzip ve Deflate algoritmaları karşılaştırılmıştır. Veri seti olarak Calgary külliyatı kullanılmıştır. Ayrıca 2,9 GHz hıza sahip, İntel Core i7 işlemcili, 8 GB Ram'li, Windows 7 Enterprise işletim sistemine sahip bir bilgisayar üzerinde test işlemleri yapılmıştır.

2.1. Sıkıştırma Yöntemleri

Yazılım kullanarak sıkıştırma

- Eğer kullanılan uygulama mimarisi uygun ise çok daha az maliyetlidir.
- Farklı mesajlaşma formatlarına göre uygun algoritmayı uygulayabilme esnekliğine sahiptir.

Örneğin, XML ve HTML için metin üzerindeki algoritmalar çok daha başarılı olurken, farklı formatlardaki (tiff, jpeg, gif v.b) banka dokümanlarında bu dokümanlar için özelleştirilmiş algoritmalar çok daha başarılı olmaktadır.

- Geliştirme ve yaygınlaştırma maliyeti düşüktür.
- Gelişen teknoloji ve yeni algoritmalar rahatlıkla uygulamaları etkilemeden devreye alınabilir.

Yazılım kullanarak sıkıştırma yapan firmalar, kullandıkları yazılım teknolojisi ile birlikte hazır gelen fonksiyon kütüphanelerini kullanabilmekte veya üçüncü parti firmaların sattığı yazılımları tercih edebilmektedirler.

Donanım kullanarak sıkıştırma

- Merkezde ve merkez ile bağlantı kuran her bir noktada donanım gerektirmektedir,
- Satın alma, işletme ve bakım maliyeti yüksektir,
- Cihazların kurulumu ve yönetimi için konusunda uzman personele ihtiyaç vardır.

2.2 Algoritma Karşılaştırması

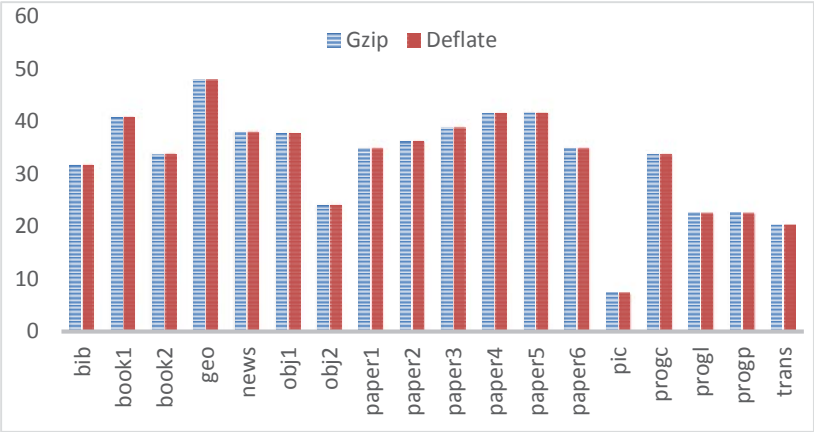
Sistemde hali hazırda .Net Framework 4.5 kütüphanesindeki GzipStream metodu kullanılmaktadır. Bu kütüphanede bayt dizilerini sıkıştırmak için Gzip algoritmasına alternatif olarak sadece DeflateStream algoritması bulunmaktadır. GzipStream metodu sıkıştırdığı veriye CRC (Cyclic Redundancy Check) bilgisi ekleyerek, transfer edilen verinin orijinaliyle birebir aynı olduğundan emin olmamızı sağlar. Bu bilgi DeflateStream metodunda bulunmamaktadır. Sisteme 3. Parti bir yazılım yerleştirmemek adına karşılaştırma için bu iki algoritma kullanılmıştır. Test verisi olarak Calgary külliyatı olarak bilinen ve veri sıkıştırma algoritmaların testlerinde

standart haline gelmiş veri seti kullanılmıştır. Bu külliyatta 13 kategoride farklı büyüklükte ve farklı sözcük sayılarına sahip veriler bulunmaktadır.

	Kategori	Boyut
bib	Bibliyografi	111261
book1	Kurgusal Yazı	768771
book2	Kurgusal Olmayan Yazı	610856
geo	Jeofiziksel Veri	102400
news	Batch File	377109
obj1	VAX Nesne kodu	21504
obj2	Apple Mac Nesne kodu	246814
paper1	Teknik Makale	53161
paper2	Teknik Makale	82199
paper3	Teknik Makale	46526
paper4	Teknik Makale	13286
paper5	Teknik Makale	11954
paper6	Teknik Makale	13314
pic	Siyah Beyaz resim	513216
progc	C Kodu	39611
progl	LISP Kodu	71646
progp	PASCAL Kodu	49379
trans	Transkiript	93695

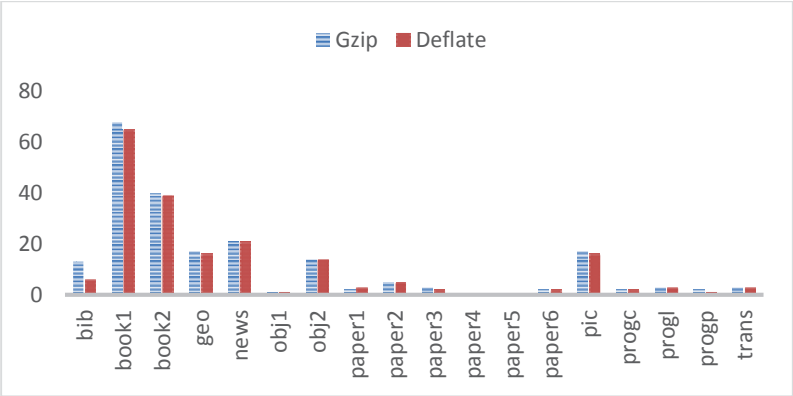
Tablo 1: Tabloda 13 kategoriye ait 18 adet veri kümesi bulunmaktadır.

Yukarıdaki veriler tek tek Gzip ve Deflate algoritmalarına uygulanarak sıkıştırma oranı (sıkıştırmadan sonraki veri boyutunun orjinal boyuta oranının 100 ile çarpılması ile elde edildi.), sıkıştırma süresi ve sıkıştırılan veriyi açma süreleri ölçüldü. Aşağıdaki şekil 1 de grafikte bu ölçüm görülmektedir.



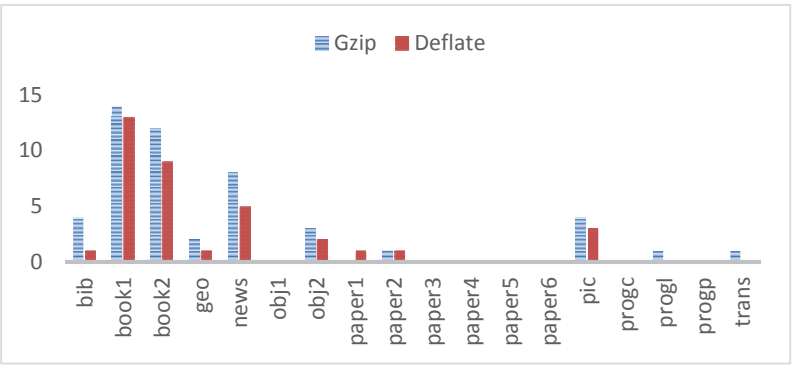
Şekil 1: Algoritmaların, 18 veri seti için sıkıştırma oranları gösterilmiştir.

Aşağıdaki şekilde(Şekil 2) ise algoritmaların sıkıştırma süreleri görülmektedir. Milisaniye cinsinden alınan ölçüm verilerinde de görüldüğü gibi sıkıştırma süreleri de neredeyse aynı çıkmıştır.



Şekil 2: Algoritmaların, 18 veri türünde sıkıştırma süreleri milisaniye cinsinden gösterilmiştir.

Aşağıdaki şekilde (Şekil 3) ise algoritmaların açma süreleri görülmektedir. Yine milisaniye cinsinden alınan veri ölçümlerinde ölçüm sonuçları birbirlerine çok yakın çıkmıştır.



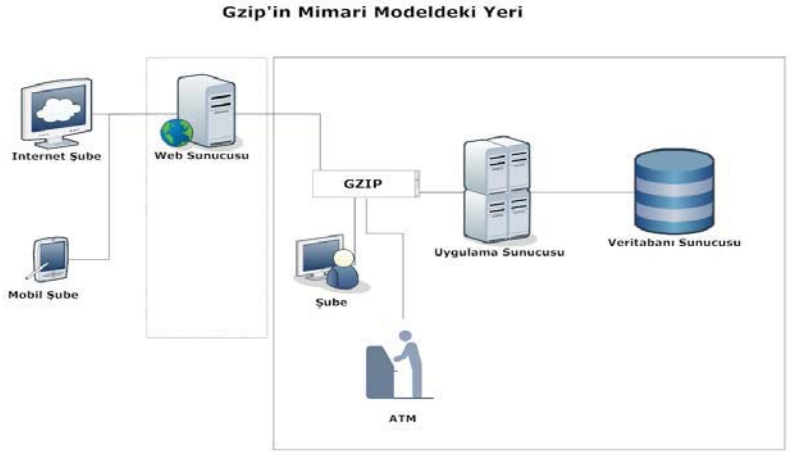
Şekil 3: Algoritmaların 18 veri türünde açma süreleri milisaniye cinsinden gösterilmiştir.

Bu banka sisteminde transfer edilen ham veri; çeşitli veri tiplerine sahip sınıfların serileştirilmesiyle oluşan bayt dizileridir. Ortalama bir şubeye gelen günlük sıkıştırılmamış veri miktarı 1124 megabayt olarak ölçülmüştür. Bu günlük veri Gzip algoritması ile sıkıştırıldığında 325 megabayta, Deflate algoritması ile sıkıştırıldığında ise 319 megabayta düşmektedir.

2.3. Finans Sektörü Uygulaması

Yaklaşık 300 şubesi bulunan bir katılım bankasında istemci uygulamalar (şube, internet şube, mobil şube, atm vb) ile uygulama sunucusu arasında veri iletişiminin daha hızlı yapılabilmesi amacıyla Gzip algoritması ile sıkıştırma yapılmaktadır. Bu çalışma da uygulanabilirlik açısından Gzip algoritmasına alternatif olarak Deflate algoritması seçilmiş ve karşılaştırma testleri yapılmıştır.

Bankanın sistemi ile her gün on binlerce kullanıcı tarafından kullanılan internet sitesi, ATM, mobil bankacılık ve şube işlemlerinin daha hızlı yapılabilmesini amaçlamaktadır. Günlük gerçekleşen 1,5 milyon işlemin her birinden kazanılan performans sistemi oldukça olumlu etkilemektedir. Kurulan sistemin bankacılık sistemine modüler bir şekilde bütünleşik olması, başka algoritma ve yöntemlerinde kullanılmasına imkân vermekte ve istenildiği ya da ihtiyaç duyulduğu takdirde yenisiyle değiştirilebilmesine olanak sağlamaktadır.



Şekil 4: Yaklaşık 300 Şubesi Bulunan Katılım Bankasında Kullanılan Blok Yapı gösterilmektedir.

Bankada, yazılım mühendisliği SOA (Service Oriented Architecture) yaklaşımlarından birisi olarak kullanılan ESB (Enterprise Service Bus) mimarisi kullanılmaktadır. Bankanın iç ve dış ağından ESB'ye ulaşan tüm mesajlar (HTTP, HTTPS, SOAP, TCP, REST) protokol bağımsız olarak uygulama katmanında Gzip algoritması aracılığı ile sıkıştırılır ve ağ üzerinde sıkıştırılmış olarak taşınır. ESB kendisine gelen tüm mesajları açarak ilgili bankacılık hizmetlerine yönlendirir ve işlem sonuçlarını aynı yol üzerinden yine sıkıştırarak cevap döner [7, 8].

Sıkıştırma yazılımının yapısında, ağ kullanımını azaltmak ve mesajlaşma hızını arttırmak amacı ile .Net Framework 4.5 kütüphanesine ait olan GzipStream sıkıştırma algoritması kullanılmaktadır. Bankada kullanılan 3 katmanlı mimarinin istemci ile uygulama sunucusu arasında gidip gelen tüm mesajları bu yapı ile sıkıştırılmaktadır. Sistemde üzerinde bulunan yaklaşık 5 milyon müşteri hesabı üzerinden gerçekleşen, günlük 1,5 milyon işlemin her birisi için yapılan sıkıştırma işlemi sistemin davranışını olumlu yönde değiştirmekte ve oldukça verimli bir hat kullanımı sağlamıştır.

Bu verimli sıkıştırmanın başarısının sonucu olarak merkezi sunucularımız ile şube istemcileri arasında 2 mega bit hat yeterli gelmektedir. Artan şube, raporlama ihtiyacı ve işlem sayıları her geçen gün ağ yoğunluğumuzu artmaktadır. Dijital çağ dolayısı ile video, ses ve görsel içeriklerin artması da bant genişliği ihtiyacını giderek artırmıştır. Bütün bunlara rağmen ülkemizde halen hat ücretlerinin çok yüksek olması bizleri sıkıştırma algoritmasının kullanılmasına ve teknolojinin takip edilerek alternatiflerin değerlendirilmesine yöneltmiştir.

3. SONUÇ

Yapılan bu çalışma sıkıştırma işleminin, web site, ATM, şubeler gibi istemcilerin merkezle aralarındaki veri trafiğini daha optimize edebilecek alternatif algoritmaları incelemek adına yapılmıştır.

Özellikle .Net Framework teknolojisi kullanan firmalar için, Gzip ve Deflate gibi sıkıştırma algoritma uygulamaları üzerinde hassasiyetle durulmuştur. Yapılan ölçümlerde hali hazırda sistemde kullanılan Gzip algoritmasına alternatif olarak düşünülecek Deflate algoritmasının ölçümleri, Gzip algoritmasının ölçümlerine çok yakın sonuçlar vermiştir. Sistemde çalışan algoritmayı değiştirme maliyeti göz önüne

alınarak ölçüm sonuçları incelenmiştir. Bu sonuçlara göre, alternatif algoritmanın kullanılmasına yetecek kadar büyük bir performans oranı sağlamadığı görülmüştür.

KAYNAKLAR

1. Altan M, "Veri Sıkıştırmada Yeni Yöntemler", PhD thesis, Institute of Natural Sciences, Trakya University, Edirne, Turkey, 2006.

2. Mesut A, Carus A, "Kayıpsız Görüntü Sıkıştırma Yöntemlerinin Karşılaştırılması", II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi, MBGAK İstanbul, p,. 93-100, 2005,

3. Enlicay M,Ülger İ, Şahin Ö, Baydarman M, Taşdemir Ş, "Bankacılık ve Finans Sektöründe Bir Veri Sıkıştırma Algoritma Uygulaması", International Conference and Exhibition on Electronic, Computer and Automation Technologies (ICEECAT'14), May 9-11, 2014

4. Ouyang J, Luo H, Wang Z, Tian J, Liu C, Sheng K, "FPGA implementation of GZIP compression and decompression for IDC services", Field-Programmable Technology (FPT), International Conference, p. 265 – 268, 2010.

5. Zhu K, Liu W, Du J, "A custom GZIP decoder for DTV application" ,Circuits and Systems (ISCAS), 2013 IEEE International Symposium , p. 681 – 684, 2013.

6. Li Y, Wang Y, "Research on Compression Technology for Geodata Based SVG in LBS, Web Information Systems and Mining (WISM)", 2010 International Conference, p. 134 – 137, 2010.

7. Pasatcha, P., "Management of Innovation and Technology", 4th IEEE International Conference, Mahanakorn Univ. of Technol., Bangkok, 1282 – 1285, 2008.

8. Available: Service-oriented architecture, http://en.wikipedia.org/wiki/Service-oriented_architecture, 04.04.2014.