

# HTTP Streaming ve Scalability

2020-03-22T13:50:03+03:00

## Contents

|   |   |
|---|---|
| Giriş . . . . .                                     | 1 |
| Ölçeklenebilirlik . . . . .                         | 2 |
| HTTP Streaming . . . . .                            | 2 |
| Su Metaforu . . . . .                               | 2 |
| Tanker Senaryosu . . . . .                          | 3 |
| Öneri . . . . .                                     | 3 |
| No Streaming ve Streaming Karşılaştırması . . . . . | 3 |
| No Streaming . . . . .                              | 4 |
| Streaming . . . . .                                 | 4 |
| Sonuç . . . . .                                     | 4 |
| Dipnotlar . . . . .                                 | 4 |

## Giriş

Günümüzde AVM ve restoranlarda nasıl ki müşteriler yemeğinin bir an önce gelmesini istiyor ve bu çoğu zaman restoran tercihinde önemli rol oynayabiliyorsa, günümüz web dünyasında da durum bundan farklı değil. Bu sebeple sunucu kaynaklarımızı dengeli ve efektif kullanmanın önemi geçen gün artmakta.

Sunucu maliyetlerimizi düşük tutmaya çalışırken bir yandan son kullanıcıya daha hızlı ve kaliteli hizmet vermek öncelikli hedeflerimizden olmalı.

Perfomance is a feature. -- Anonim

HTTP Streaming yöntemi, dolaylı yoldan sunucu maliyetini azaltırken, son kullanıcıya daha hızlı hizmet vermeyi sağlar.

## Ölçeklenebilirlik

Scalability <sup>1</sup> is the property of a system to handle a growing amount of work by adding resources to the system. In an economic context, a scalable business model implies that a company can increase sales given increased resources. -- *Wikipedia*

Türkçe olarak:

Bir sistemin artan yükün altından kaynak ekleyerek kalkma becerisidir. Ekonomik bağlamda ise yatırım karşılığında gelirin artabilmesidir.

İdeal şartlarda bir birim kaynak artırımını ile ek bir birim daha iş yapılması beklenir. <sup>2</sup>

## HTTP Streaming

Giriş bölümünde "*Eşdeğer bir kaynakla daha fazla kullanıcıya hizmet vermek*" ifadesini kullandık.

*Stream* <sup>3</sup> kelimesinden yola çıkalım:

A steady current in such a flow of water.

Türkçe'ye ifade edecek olursak:

*Sabit bir su akışı* <sup>4</sup>

## Su Metaforu

Köyün yakınındaki bir su pınarı var ve köye su lazım. İki yolumuz var

1. Tankerle su getirmek ve köy halkının bidonlarını tankerin ana hortumu ile doldurmalarını sağlamak,
2. Tankerin ana hortumuna çok sayıda hortum bağlayarak aynı anda olduğunu bildiğince fazla sayıda kişiye su sağlamak.

Burada

- Tankerimiz, sunucularımızın Disk/Network <sup>5</sup> veya RAM kapasitesini
- Bidon ve varillerin boyutu ise kullanıcıların web uygulamasından talep ettiği dosya (*response*) boyutunu,

<sup>1</sup><https://en.wikipedia.org/wiki/Scalability>

<sup>2</sup>Bu 1/1 oranının ideal şartlarda bile 1'den düşük olduğu "*Carno Teoremi*"([https://en.wikipedia.org/wiki/Carnot's\\_theorem\\_\(thermodynamics\)](https://en.wikipedia.org/wiki/Carnot's_theorem_(thermodynamics)))" ve "*Termal Verimlilik*" prensipleri ile tanımlanmıştır. [https://en.wikipedia.org/wiki/Thermal\\_efficiency](https://en.wikipedia.org/wiki/Thermal_efficiency)

<sup>3</sup><https://www.wordnik.com/words/stream>

<sup>4</sup><https://en.wikipedia.org/wiki/Streaming>

<sup>5</sup><https://en.wikipedia.org/wiki/Input/output>

- Köy halkının sıraya girdikten sonra tankerden su alana kadarki süresi ise web sunucusunun kullanıcının isteğini tamamlama süresini

temsil edebilir. Bu yazımızda kaynak olarak odak noktamız RAM olacak.

## Tanker Senaryosu

Tanker senaryosunda, köy sakinleri su ihtiyaçları için pınara bir tanker gönderir ve bütün köy halkı tankerin gelmesini bekler.

Tanker geldikten sonra sıraya girerler, tankerin hortumu yardımıyla kovalarını doldurduktan sonra evlerine dönerler.

Köy sakinleri suya ulaşmak için:

- Tankerin gidip dönmesini,
- Tanker geldikten sonra sıranın kendileri gelmesini,
- Büyük hacimli bidon ve varillerle gelen kişilerin su doldurmalarını

beklediler. Çünkü tankerin bir adet tahliye hortumu var ve **aynı anda sadece bir kişiye** hizmet verebiliyor.

Peki sıranın en önündeki kişi **bir tanker** suyu almak isterse ne olacak?

- İkinci sıradaki kişi dahil bir tanker suyun boşalmasını bekleyecek,
- Tanker, su doldurmak için tekrar pınara gidecek.

## Öneri

{{% notice tip %}} Mümkünse sıradaki kişi sayısı kadar küçük hortumu tankerimizin ana hortumuna bağlayarak sıradaki bütün kişilerin bekleme sürelerini dengeleyerek bütün köy halkını memnun edebiliriz. {{% /notice %}}

Böylece büyük varillerle gelen kişilere de aynı kapasitede debi sağlanmış ve kaynak dengesizliğinin önüne önemli ölçüde geçilmiş oldu.

Buradaki önemli noktalar:

1. Tankerin kapasitesi
2. Ana hortuma bağlayabileceğimiz küçük hortum adeti.
3. Yeni hortumlar ekleyerek daha fazla kişiye hizmet verebilmek kısacık: **Ölçeklenebilirlik**<sup>6</sup>

## No Streaming ve Streaming Karşılaştırması

Metaforumuzu yeterince açıkladıktan sonra bunu web uygulamamıza uyarlayalım.

<sup>6</sup><https://en.wikipedia.org/wiki/Scalability>

**1024 MB RAM** kapasiteli bir sunucumuz ve kullanıcılarımız bizden **512MB** boyutlu farklı dosyalar istemiş olsun.

### No Streaming

- Bir ve ikinci kullanıcının dosyalarını disk veya networkten okuyarak RAM'e yazmayı **bekledik**
- Üçüncü kullanıcımız ise ilk iki kullanıcının isteklerinin tamamlanmasını **bekleyecek** çünkü yeni bir **512MB** dosyayı yazmak için RAM kalmadı.
- **512MB** boyutlu dosyaların RAM'den boşaltılması **zaman alacak** çünkü ne kullanıcılarımızın internet kapasitesi ne de network kapasitemiz bu dosyaları bir anda kullanıcıya ulaştırmak için yeterli değil.
- Üçüncü kullanıcımız **hâlâ sırada bekliyor**.

### Streaming

Streaming senaryosunda ise bir yandan disk veya networkten okurken, diğer yandan bu okuduğumuz veriyi kullanıcılarımıza transfer ederiz.

- **1MB** parçalar halinde dosyalarımızı okuduğumuzu
- Diskimizin saniyede **32MB** dosya okuma kapasitesi olduğunu
- **Network** kapasitemizin aynı anda **32** kişiye **1MB** dosya ulaştırabildiğini ve kullanıcılarımızın internet kapasitelerinin saniyede en az **1MB** indirebildiğini farz edelim.
- Sunucumuz 32 kişiyi sıraya aldı ve kullanıcılara ait dosyaları diskten okuyarak kullanıcılara ulaştırmaya başladı.
- Her 512 saniyede 32 kişinin isteklerini sonlandırarak yeni kullanıcılara hizmet vermeye başlayabilecek.

### Sonuç

Bu yazımızda "Streaming" yöntemini üzerinden anlatmaya çalıştım. Sonraki yazılarımızda uygulamalı olarak bu konulara değinmeye çalışacağım.

- ASP.Net Core ile JSON ve File Streaming
- SQL Server üzerinden Streaming ile veri çekmek
- HTTP Client ile streaming download
- Son kullanıcıya yansımaları
- Performans etkileri

konularından bahsedeceğiz.

### Dipnotlar