

Yazılım Test Otomasyon Araçlarının Hız Boyutunda Karşılaştırılması

Ahmet Mert AVUNDUK, Haluk YILDIZ, Mustafa Kemal TUNA

Anahtar Kelimeler: Selenium, NodeJS, Fonksiyonel Test Otomasyonu, CasperJS, PuppeteerJS, Hız Testi

Özet: Fonksiyonel test, fonksiyonel yazılım gereksinimlerinin sınanması için kullanılan bir yöntemdir. Fonksiyonel test otomasyon araçları ise fonksiyonel test senaryolarını; kod yardımıyla tanımlayıp, otomatik olarak çalıştırıp, sonuçları raporlayan araçlardır. Selenium, CasperJS ve Puppeteer, NodeJS ortamında sıkça kullanılan fonksiyonel test otomasyon araçlarıdır. Bu çalışma bahsi geçen fonksiyonel test otomasyon araçlarının hız testi karşılaştırmalarını içerecektir.

1. Giriş

"Fonksiyonel test otomasyonu: bir yazılımın, diğer bir yazılımın test senaryosunun yürütülmesinin kontrolüdür"[1] Selenium, CasperJS ve Puppeteer araçları üzerinde, NodeJS ortamında, farklı web uygulamaları üzerindeki form işlemleri için Javascript aracılığı ile test senaryoları yazılacak ve testlerin yürütme süresi temelinde görsel ve nümerik analiz yapılp sonuçlar sunulacak ve oluşan farklılıkların kök sebep analizi çalışması için her bir araç üzerinde Javascript'in çalışma süresi, HTML parsing süresi, CSS parsing süresi karşılaştırılacaktır.

2. İlgili Çalışmalar

"Performance Evaluation & Comparison of Software Testing Tool"[1] isimli makalede Selenium ve QTP test otomasyon araçları için benchmark testi yapılarak, iki test otomasyon aracı karşılaştırılmıştır. Çevrimiçi bir internet sitesinin üzerinde her iki test aracı ile yapılan testlerin karışıklık ve çalışma zamanı kaydedilerek karşılaştırma yapılmıştır. Sonuç olarak Selenium ilgili otomasyon testini 42 saniye de gerçekleştirirken QTP 58 saniye de gerçekleştirmiştir.

"Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete"[2] isimli makalede Selenium , QTP ve Testcomplete test otomasyon araçlarının karşılaştırılması özelliklerine göre yapılmıştır. Sonuç olarak en iyi test aracının QTP olduğu sonucuna ulaşılmıştır.

3. Test Otomasyon Araçlarına Genel Bakış

3.1 Selenium

“Selenium’u tanımlayacak olursak tarayıcı otomasyon aracı diyebiliriz. Yani web siteniz üzerindeki bazı adımların sizin belirleyeceğiniz şekilde tarayıcı üzerinde otomatik test eden bir araç. Örneğin web sitenizde bir kullanıcı giriş alanı var ve bunun çalışıp çalışmadığını istediğiniz zaman test etmek istiyorsunuz. Kullanıcı giriş sayfası için test etmek istediğiniz adımları oluşturup istediğiniz zaman tek bir komutla bunların testini yapıp size bir rapor sunmasını sağlayabilirsiniz.”[4]

Selenium; Java, C#, Python, Ruby, PHP, Perl ve Javascript dilleriyle kullanılabilir. Selenium bu diller dışında kendi IDE’ sinin sağladığı kaydetme ve yeniden oynatma özellikleriyle de kullanılabilir.

3.2 CasperJS

Casperjs, Phantomjs ve Slimerjs üzerine Javascript kullanılarak geliştirilmiş bir test kütüphanesidir.

“Casperjs ile test senaryoları oluşturarak belirlediğimiz parametreler ile bu testlerin uygulanmasını sağlayabiliyoruz. Örneğin geliştirmekte olduğumuz bir kullanıcı sistemi olduğunu varsayalım. Tarayıcı üzerinde test ederken input a kullanıcı adı ve parolayı girmek, submit butonuna basmak yada çıkış yapmak gibi tüm işlemleri komut olarak yazıyor ve test edebiliyoruz. Test esnasında istediğimiz aşamada ekran görüntüsü alabiliyoruz.”[2]

3.3 Puppeteer

“Puppeteer, google tarafından geliştirilen chrome’u headless (herhangi bir kullanıcı arayüzü olmadan) olarak yönetebileceğiniz api sunan bir nodejs kütüphanesidir.”[3] (pptr.dev)

Puppeteer; sayfaların ekran görüntüsünü ve PDF’ ini üretebilmektedir. Kullanıcı arayüzünü test edebilmekle beraber klavye girdilerini ve form gönderme işlevlerini otomatize edebilmektedir.

4. Test Ortamı ve Ortamın Hazırlanması

Selenium, Casperjs ve Puppeteer otomasyon test araçlarının hız performanslarının karşılaştırılabilmesi amacıyla iki farklı test ortamında testler koşturulacaktır. Bunlar çevrimiçi bir internet sitesi ve yerel sunucu ortamı içinde gerçekleştirilecektir. Yerel sunucuda hazırlanan siteler çevrimiçi sitenin kaynak kodundan script ve iframe etiketlerinin silinerek sayfaların çevrimiçi bağımlılığı azaltılmıştır. Test ortamı için seçilen web sayfaları ;

Sayfa1: www.calculator.com/calculate/loan/

Sayfa2: www.calculator.com/calculate/feet_inches/

Sayfa3: www.calculator.com/calculate/mortgage/

Sayfa1 Yerel: Sayfa 1'in yerel sunucuya taşınmış halidir.

Sayfa2 Yerel: Sayfa 2'in yerel sunucuya taşınmış halidir.

Sayfa3 Yerel: Sayfa3'in yerel sunucuya taşınmış halidir.

4.1 Aynı site de farklı sayfalarda testlerin koşturulması

“www.calculator.com” web sitesi içerisinde bulunan farklı sayfalar üzerinde testler yapılarak, ilgili test otomasyon araçlarının test karşılaştırması yapılacaktır. Bu site içerisinde Sayfa1, Sayfa2 ve Sayfa3 sayfalarında otomasyon testleri koşturulacaktır. Burada amaç aynı sunucu içinde testler arasında oluşacak benchmark sonuçlarının serverdan bağımsız elde edilmesini sağlamaktır.

4.1.1 Test Senaryoları

Senaryo No	Test Aracı	Test Ortamı	Aşamalar
1	Selenium	Sayfa2	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Length1 değerini gir.4. Length2 değerini gir.5. Feet1 değerini gir .6. Feet2 değerini gir.7. Calculate butonuna bas.8. Sonucu doğruluğunu karşılaştır.
2	CasperJS	Sayfa2	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Length1 değerini gir.4. Length2 değerini gir.5. Feet1 değerini gir .6. Feet2 değerini gir.7. Calculate butonuna bas.8. Sonucu doğruluğunu karşılaştır.
3	Puppeteer	Sayfa2	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Length1 değerini gir.4. Length2 değerini gir.5. Feet1 değerini gir .6. Feet2 değerini gir.7. Calculate butonuna bas.8. Sonucu doğruluğunu karşılaştır.
4	Selenium	Sayfa1	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Loan Amount değerini gir.4. Annual Interest değerini gir.5. Payment Each Period değerini gir .6. Periods per Year değerini gir.7. Total Number of Payments değerini gir.8. Total Number of Years değerini gir.9. Calculate butonuna bas.

			10. Sonucu doğruluğunu karşılaştır.
5	CasperJS	Sayfa1	1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Loan Amount değerini gir. 4. Annual Interest değerini gir. 5. Payment Each Period değerini gir . 6. Periods per Year değerini gir. 7. Total Number of Payments değerini gir. 8. Total Number of Years değerini gir. 9. Calculate butonuna bas. 10. Sonucu doğruluğunu karşılaştır.
6	Puppeteer	Sayfa1	1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Loan Amount değerini gir. 4. Annual Interest değerini gir. 5. Payment Each Period değerini gir . 6. Periods per Year değerini gir. 7. Total Number of Payments değerini gir. 8. Total Number of Years değerini gir. 9. Calculate butonuna bas. 10. Sonucu doğruluğunu karşılaştır.
7	Selenium	Sayfa3	1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Mortgage Amount değerini gir. 4. Interest Rate (%) değerini gir. 5. Starting date değerini gir . 6. Prepayment Amount değerini gir. 7. Calculate butonuna bas. 8. Sonucu doğruluğunu karşılaştır.
8	CasperJS	Sayfa3	1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Mortgage Amount değerini gir. 4. Interest Rate (%) değerini gir. 5. Starting date değerini gir . 6. Prepayment Amount değerini gir. 7. Calculate butonuna bas. 8. Sonucu doğruluğunu karşılaştır.
9	Puppeteer	Sayfa3	1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Mortgage Amount değerini gir. 4. Interest Rate (%) değerini gir. 5. Starting date değerini gir . 6. Prepayment Amount değerini gir. 7. Calculate butonuna bas. 8. Sonucu doğruluğunu karşılaştır.

4.2 Yerel sunucuda farklı sayfalarda testlerin kořturulması

Yerel sunucuda Sayfa4, Sayfa5 ve Sayfa6 için ayrı ayrı otomasyon testleri kořturulacaktır. Burada amaç sunucu etkilerinden bağımsız olarak benzer fonksiyon özelliklerine sahip yazılımların otomasyon testleri arasında benchmark sonuçlarının elde edilmesini sağılamaktır.

4.2.1 Test Senaryoları

Test senaryosu çalıştırılmadan önce yerel sunucu çalıştırılıp testler yerel sunucu üzerinde kořulacaktır.

Senaryo No	Test Aracı	Test Ortamı	Ařamalar
1	Selenium	Sayfa2 Yerel	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Length1 deęerini gir.4. Length2 deęerini gir.5. Feet1 deęerini gir .6. Feet2 deęerini gir.7. Calculate butonuna bas.8. Sonucu doęruluęunu karřılařtır.
2	CasperJS	Sayfa2 Yerel	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Length1 deęerini gir.4. Length2 deęerini gir.5. Feet1 deęerini gir .6. Feet2 deęerini gir.7. Calculate butonuna bas.8. Sonucu doęruluęunu karřılařtır.
3	Puppeteer	Sayfa2 Yerel	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Length1 deęerini gir.4. Length2 deęerini gir.5. Feet1 deęerini gir .6. Feet2 deęerini gir.7. Calculate butonuna bas.8. Sonucu doęruluęunu karřılařtır.
4	Selenium	Sayfa1 Yerel	<ol style="list-style-type: none">1. Browser'ı aç2. Sayfa elemanlarının oluşmasını bekle3. Loan Amount deęerini gir.4. Annual Interest deęerini gir.5. Payment Each Period deęerini gir .6. Periods per Year deęerini gir.7. Total Number of Payments deęerini gir.8. Total Number of Years deęerini gir.9. Calculate butonuna bas.10. Sonucu doęruluęunu karřılařtır.

5	CasperJS	Sayfa1 Yerel	<ol style="list-style-type: none"> 1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Loan Amount değerini gir. 4. Annual Interest değerini gir. 5. Payment Each Period değerini gir . 6. Periods per Year değerini gir. 7. Total Number of Payments değerini gir. 8. Total Number of Years değerini gir. 9. Calculate butonuna bas. 10. Sonucu doğruluğunu karşılaştır.
6	Puppeteer	Sayfa1 Yerel	<ol style="list-style-type: none"> 1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Loan Amount değerini gir. 4. Annual Interest değerini gir. 5. Payment Each Period değerini gir . 6. Periods per Year değerini gir. 7. Total Number of Payments değerini gir. 8. Total Number of Years değerini gir. 9. Calculate butonuna bas. 10. Sonucu doğruluğunu karşılaştır.
7	Selenium	Sayfa3 Yerel	<ol style="list-style-type: none"> 1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Mortgage Amount değerini gir. 4. Interest Rate (%) değerini gir. 5. Starting date değerini gir . 6. Prepayment Amount değerini gir. 7. Calculate butonuna bas. 8. Sonucu doğruluğunu karşılaştır.
8	CasperJS	Sayfa3 Yerel	<ol style="list-style-type: none"> 1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Mortgage Amount değerini gir. 4. Interest Rate (%) değerini gir. 5. Starting date değerini gir . 6. Prepayment Amount değerini gir. 7. Calculate butonuna bas. 8. Sonucu doğruluğunu karşılaştır.
9	Puppeteer	Sayfa3 Yerel	<ol style="list-style-type: none"> 1. Browser'ı aç 2. Sayfa elemanlarının oluşmasını bekle 3. Mortgage Amount değerini gir. 4. Interest Rate (%) değerini gir. 5. Starting date değerini gir . 6. Prepayment Amount değerini gir. 7. Calculate butonuna bas. 8. Sonucu doğruluğunu karşılaştır.

4.3 Karşılaştırma Sonuç Raporunun Hazırlanması

Test senaryolarının her biri için benchmark değerleri tablo haline getirilerek, oluşturulan tablolardan görsel grafikler oluşturularak kullanılan üç test aracının performans analizi yapılmıştır.

5. Analiz Ve Bulguların Yorumlanması

5.1 Analize Giriş

Bu aktiviteye başlamak için gereken girdiler şunlardır:

- * *Yerel sunucu ortamının hazırlanması*
- * *NodeJS ortamının kurulması*
- * *CasperJS ortamının kurulması*
- * *Test edilecek web sayfaları için Selenium, Puppeteer ve CasperJS kodlarının hazırlanması*

5.2 Aktiviteler

5.2.1 Test Senaryolarının Yürütülme Zamanlarının Toplanabilmesi İçin Testlerin Koşturulması

Yürütülme zaman verilerinin toplanması için test senaryolarında belirtilen her bir sayfa için test araçlarını yüzer defa çalıştırılarak, her bir çalışmanın süresi veri olarak kaydedilmektedir.

5.2.2 Verilerin Analiz Edilmesi

Her bir test aracı için bulunan elli değerlerin ortalama değerleri hesaplanacak ve bu değerler karşılaştırılacaktır. Bu karşılaştırma sonucunda kullanılan otomasyon test araçlarının hızları hakkında analiz yapılacaktır.

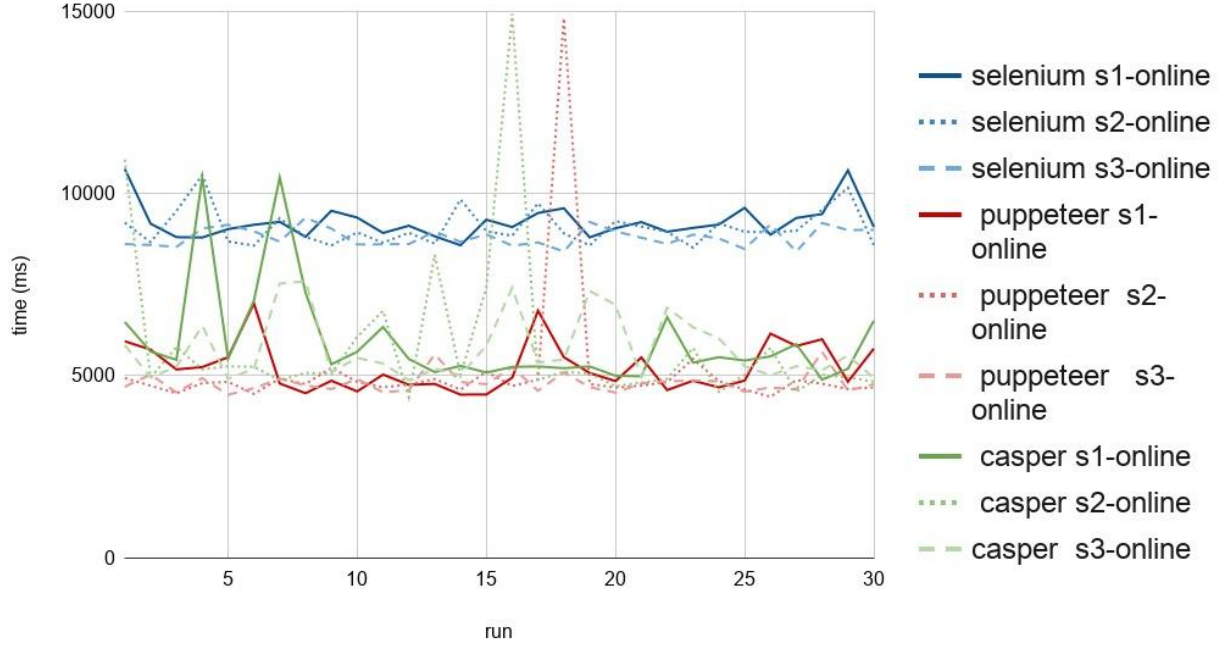
5.2.3 Analiz Çıktıları

- * *Çevrimiçi web sitesi ortamında otomasyon test araçlarının karşılaştırma grafiği*
- * *Yerel sunucu ortamında otomasyon test araçlarının karşılaştırma grafiği*

5.3 Grafiklerin Yorumlanması

5.3.1 Çevrimiçi ve Yerel Sunucu Gerçek Ortam Hız Testleri

Çevrimiçi Sunucu Gerçek Ortam Hız Testi

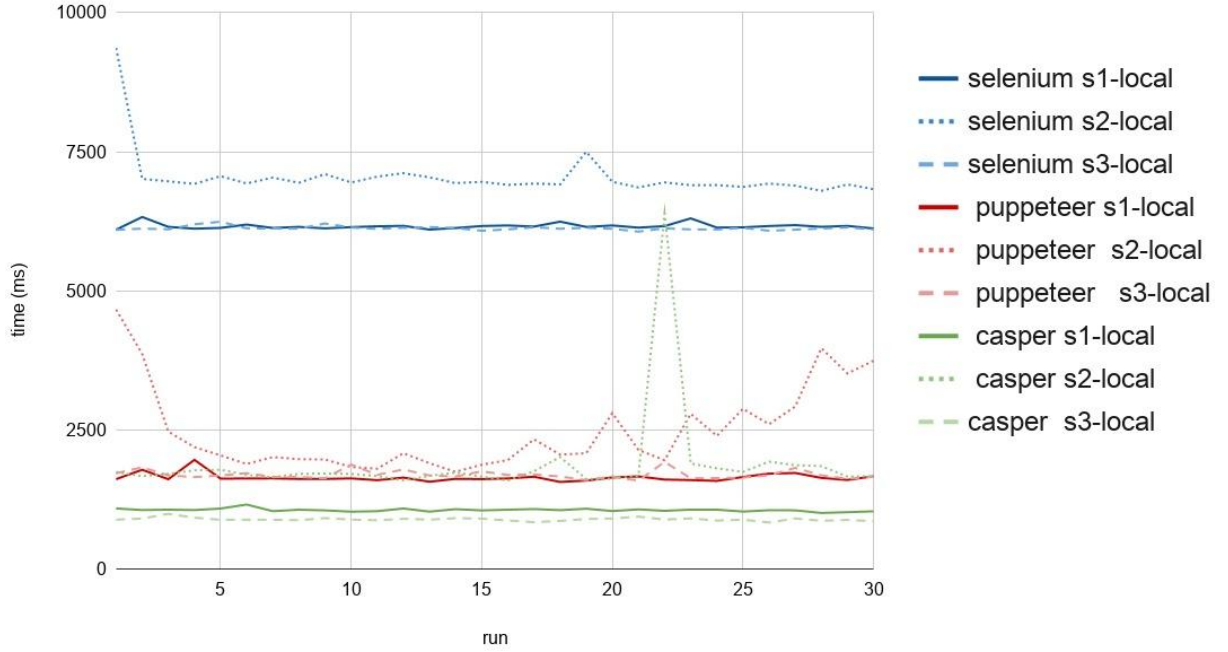


Grafik 1

Javascript Çalıştırma Hızı Ortalama Değerler		
Selenium	Puppeteer	CasperJs
En Yavaş(9036ms)	En Hızlı(5059ms)	Ortanca(5845ms)

Tablo 1

Yerel Sunucu Gerçek Ortam Hız Testi



Grafik 2

Javascript Çalıştırma Hızı Ortalama Değerler		
Selenium	Puppeteer	CasperJs
En Yavaş(6445ms)	Ortanca(1942ms)	En Hızlı(1281ms)

Tablo 2

Self Time	Total Time	Activity
43.6 ms 11.9 %	113.2 ms 30.9 %	Parse HTML
4.9 ms 1.3 %	88.1 ms 24.0 %	Evaluate Script
10.7 ms 2.9 %	80.1 ms 21.8 %	Event: mouseover
67.0 ms 18.3 %	69.9 ms 19.0 %	Function Call
55.2 ms 15.1 %	55.2 ms 15.1 %	Recalculate Style
40.2 ms 11.0 %	40.2 ms 11.0 %	Compile Script
29.4 ms 8.0 %	29.4 ms 8.0 %	Parse Stylesheet

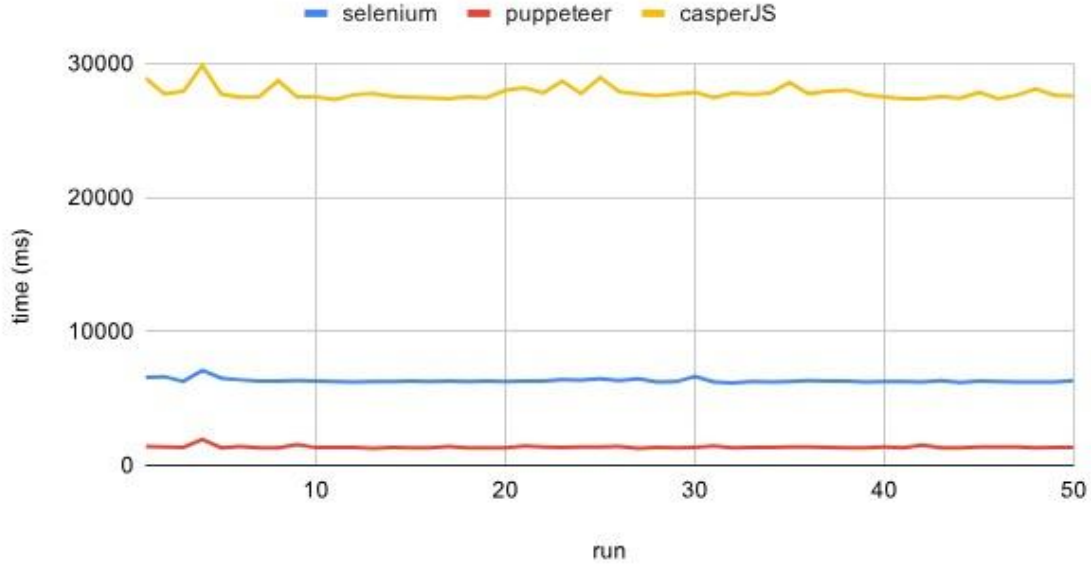
Figure 1

- Grafik 1 ve Grafik 2 de görüldüğü gibi üç farklı test otomasyon aracı aynı web uygulamasını farklı sürelerde testlerini gerçekleştirmektedir. Bunun alt nedenlerini inceleyecek olursak; Figure 1’de test otomasyonu esnasında aktivite zamanları incelendiğinde her bir test aracının farklı zamanlarda fonksiyonel testlerini tamamlaması HTML parsing, Javascript çalıştırma ve CSS parsing zamanının farklı olmasından kaynaklandığı düşünüldü. Bu nedenle her bir test otomasyon aracı için Javascript çalıştırma, HTML parsing ve CSS Parsing süreleri ayrı ayrı incelendi.

5.3.2 Javascript Çalıştırma Hız Testleri

Her bir test aracı için hazırlamış olduğumuz belli bir sayıya kadar olan asal sayıyı bulan Javascript kodu ile 50 defa yürüttüğümüzde Grafik 3 teki değerleri görmekteyiz.

Javascript çalıştırma hızı karşılaştırması



Grafik 3

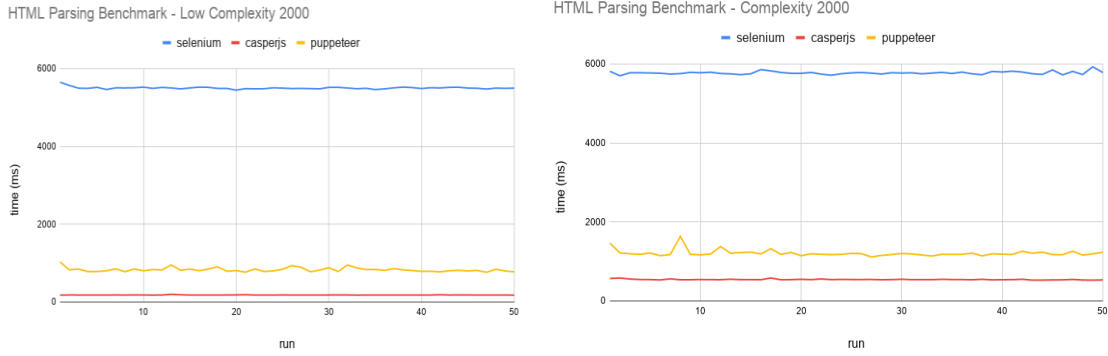
Javascript Çalıştırma Hızı Ortalama Değerler		
Selenium	Puppeteer	CasperJs
6312,08 ms	1359,82 ms	27794,74 ms

Tablo 3

Grafik 3 ve Tablo 3 de görüldüğü üzere Javascript çalıştırma hızlarına göre CasperJs'in çok yavaş Puppeteer'in ise çok hızlı olduğu görülmüştür. Çevrimiçi test koşullarında(Tablo 1) Puppeteer'in CasperJs ten Local de ki testlere(Tablo 2) göre daha hızlı olmasının sebebinin çevrimiçi sayfaların javascript kodları temizlenmediği için Javascript yoğun iş gücü gerektirip test sonuçlarında Puppeteer'i öne geçirmiştir.

5.3.2 HTML Parsing Hız Testleri

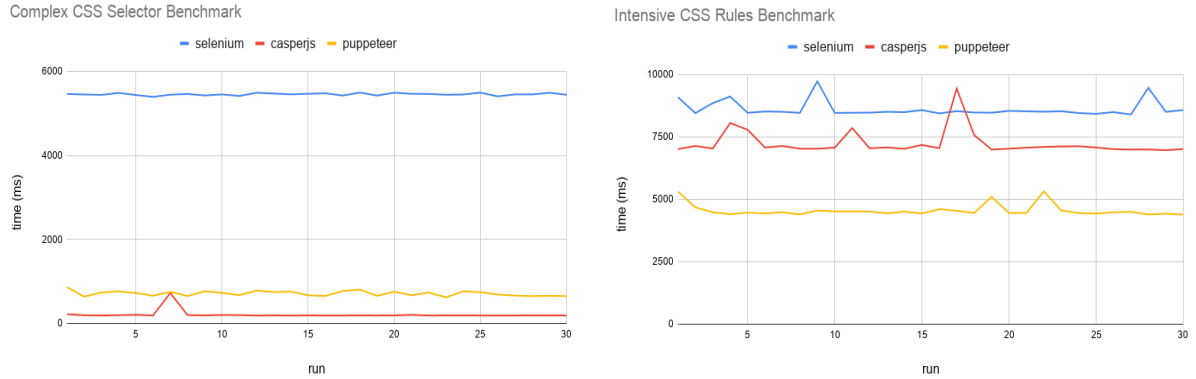
Her bir test aracı için hazırlamış olduğumuz yüksek karmaşıklık ve düşük karmaşıklık seviyelerinde sadece Html yoğunluklu kodumuzda testlerimizi 50 defa yürüttüğümüzde Grafik 4 teki değerleri görmekteyiz.



Grafik 4

5.3.3 CSS Parsing Hız Testleri

Her bir test aracı için hazırlamış olduğumuz yüksek karmaşıklık ve düşük karmaşıklık seviyelerinde sadece CSS yoğunluklu kodumuzda testlerimizi 50 defa yürüttüğümüzde Grafik 5 teki değerleri görmekteyiz.

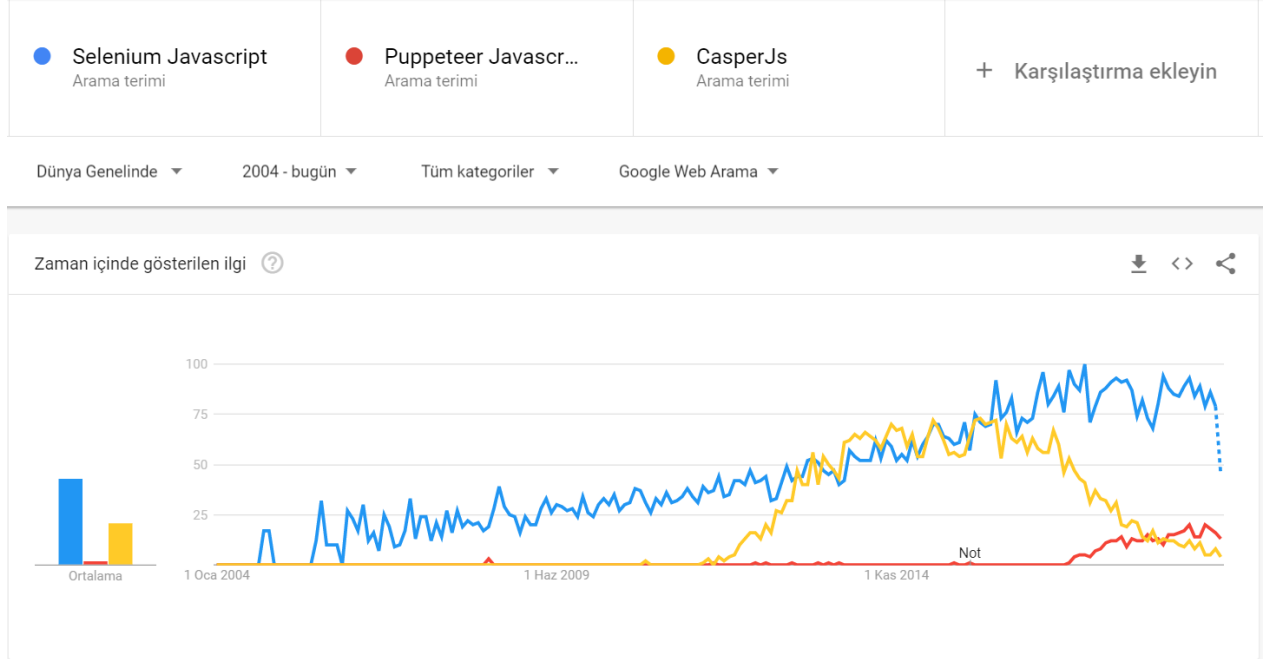


Grafik 5

5.4 Nitelik Karşılaştırması

Nitelikler	Selenium	Puppeteer	CasperJS
Çevrimiçi Test Yürütme Hızı	En Yavaş(9036ms)	En Hızlı(5059ms)	Ortanca(5845ms)
Yerel Test Yürütme Hızı	En Yavaş(6445ms)	Ortanca(1942ms)	En Hızlı(1281ms)
HTML parsing Hızı			
Javascript Çalıştırma Hızı	Ortanca	En Hızlı	En Yavaş
CSS Çalıştırma Hızı			
Ekran Görüntüsü Sunma Hızı			
Platform	Cross Platform	Cross Platform	Cross Platform
Javascript Motoru	Cromium V8	Cromium V8	WebKit
Son Yayınlanma Tarihi	09/2019	11/2019	11/2017
Maliyet	Açık Kaynak	Açık Kaynak	Açık Kaynak
Destek	Selenium Sponsorları	Google	Yok

Yıllara göre test araçlarının popülerliği



Grafik 5

6. Sonuç

Yapılan testler sonucunda:

- Çevrimiçi sunucu üzerinde yapılan gerçek ortam testlerinde CasperJs ve Puppeteer'in yaklaşık değerler verdiği, Selenium'un ise görece daha yavaş çalıştığı gözlenmiştir. Online ortamda çalıştırılan testlerde istikrarlı olmayan sonuçlar görülmüş olsa bile 30 kez yapılan tekrarla ortalama değerler elde edilmiştir.
- Yerel sunucu üzerinde yapılan testlerde CasperJS en hızlı, Selenium ise en yavaş olarak gözlemlenmiştir. Puppeteer'den elde edilen sonuçlar ise bu değerlerin arasındadır. Yerel sunucu ortamında yapılan testlerde çevrimiçi testlere göre çok daha stabil değerler alınmıştır.
- Test araçlarının farklı sürelerde testlerini gerçekleştirme kök sebeplerine baktığımızda CasperJS test otomasyon aracının HTML parsing hızının diğer araçlardan iyi olması ve HTML parsing süresinin test durumlarımızın daha HTML yoğunluklu siteler olması nedeniyle CasperJS değerleri daha iyi sonuçlar vermiştir.

Sonuç olarak bu çalışmadan elde ettiğimiz veriler doğrultusunda test edeceğimiz web sitesi Javascript ve CSS ağırlıklı bir web uygulaması ise Puppeteer daha iyi bir performans sunarken HTML ağırlıklı bir web sitesinin testinde CasperJS daha iyi bir performans sağlayacaktır.

Kaynaklar

[1]R. Rattan¹, Shallu², Performance evaluation comparison of software testing tool, International Journal of Information and Computation Technology 3 (2013) 711–716.

[2] Performance Testing Tools: A Comparative Study IJISSET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 4, June 2014. ISSN 2348 – 7968

[3] A Method for Measuring the Structure Complexity of Web Application, Wuhan University Journal of Natural Sciences Article ID: 1007-1202(2006)01-0143-08