

Java, Python ve Ruby Dillerinin Performans Karşılaştırması

Mustafa Şahin

Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği Bölümü, 17100, Çanakkale
msahin@comu.edu.tr

Özet: Popülerlikleri her geçen gün artan Java, Python ve Ruby programlama dillerinin, çeşitli test uygulamalarındaki çalışma süreleri, bellek tüketimi ve kod uzunlukları konularında karşılaştırma deneyleri yapılmış ve sonuçlar verilmiştir. Çalışma süreleri testlerinde ortalamada Java Python'a göre 7.5 kat; Ruby'ye göre 21 kat daha hızlı çalışmaktadır. Ruby ve Python kod uzunlukları ve yazım süreleri incelendiğinde Java dilinin yarısı kadar uzunluğa sahiptir. Ruby en az, Python ise en çok bellek tüketimine sahiptir.

Anahtar Kelimeler: Java, Python, Ruby, Performans.

Abstract: Popularities of Java, Python and Ruby programming languages are increasing every-day. In this work, benchmarks of runtime, memory consumption, code length were done and results are given. On average Java is 7.5 times faster than python and 21 times faster than Ruby at runtime benchmarks; Code length and developing time of Ruby and Python have half of Java. Memory consumption is the smallest for Ruby and Python is the largest.

Keywords: Java, Python, Ruby, Performance.

1. Giriş

Java geniş çevrelerce kabul görmüş, işletim sisteminden bağımsız, taşınabilir programlar yazmak için geliştirilmiş, nesneye yönelik bir programlama dilidir[1]. Python[2] ve Ruby[3] de açık kaynak fikrini ve Linux işletim sistemini destekleyen çevrelerce kabul görmüş, yıldızları her geçen gün daha fazla parlayan nesneye yönelik programlama dilleridir[4]. Her üç dil de yorumlanarak çalıştırılırlar. Farklı olarak Java programları yorumlanmadan önce byte kod adı verilen bir ara dile derlenirler. Son zamanlarda açık kaynak yazılım fikrinin hızlı yükselişi sonucunda her üç programlama dilinin gelişimi hızlanmış ve yeni sürümleri çıkarılmıştır. Bu çalışmamda bu üç dilin farklı on bir test uygulaması ile performansları incelenmiş ve testlerin sonuçları verilmiştir.

2. Testler

Dillerin performans karşılaştırma çalışmaları daha önce de gerçekleştirilmiştir[5][6][7], ça-

lışmalarda programlama dillerinin eski sürümleri kullanılmıştır ve istenilen üç dilin tamamı karşılaştırmalara alınmamıştır.

Yapılan testlerde kullanılan uygulamalar aynı ihtiyaçlara cevap veren uygulamalardır. Testler her uygulama için beşer kez gerçekleştirilmiş ve sonuç olarak ortalama değerler kullanılmıştır.

Yapılan testler ile üç programlama dilinin çalışma süreleri, bellek tüketimleri ve yazılan kod uzunlukları incelenmiştir.

2.1. Çalışma Süreleri

Karşılaştırma testleri için kullanılan uygulamalar Tablo 1'de açıklanmıştır. Bu testler her dil için beşer kez hem Linux hem de Windows işletim sistemlerinde çalıştırılmış ve ortalama sonuçlar kullanılmıştır. Çalışma süreleri için sonuçlar; Windows işletim sistemi için Tablo 2'de saniye olarak, Şekil 1'de yüzdeler olarak; Linux işletim sistemi için Tablo 3'te saniye olarak, Şekil 2'de yüzdeler olarak verilmiştir.

2.2. BellekTüketimi

Karşılaştırma testlerinde kullanılan uygulamaların bellek tüketimleri [8],[9],[10] ve [11]'de ki araçlar kullanılarak belirlenmiş ve yüzde olarak Tablo 4 ve Şekil 3 'te verilmiştir.

2.3. Yazılan Kod Uzunluğu

Uygulamaları karşılaştırmak için satır sayıları ve testlerde kullanılan harf sayıları veri olarak kullanılmıştır. Sonuçlar Şekil 4 ve Şekil 5'te verilmiştir.

3. Test Platformu

Aşağıdaki bilgisayar testler için kullanılmıştır;

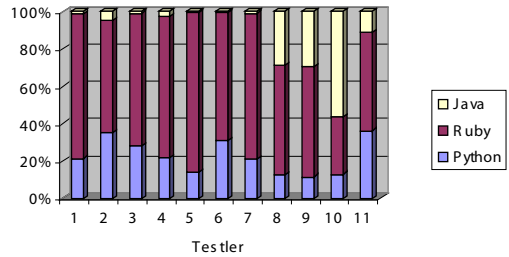
- HP Compaq NX9005 serisi Notebook
- 30 GB Harddisk
- 1 GB RAM
- Mobile AMD Athlon XP2400+ işlemci
- Windows 2003, Debian 3.1 Linux İşletim Sistemi
- Java 6.0, Python 2.5, Ruby 1.8.5

1	[0..3000000] aralığındaki sayıların ortalamasının hesaplanması.
2	[2..50000000] aralığındaki asal sayıların hesaplanması.
3	[1..100000000] aralığında çalışan boş döngü.
4	500x500 boyutlarında iki matrisin çarpımı.
5	10000 elemanlı, eleman değerleri en kötü durumda olan dizinin Kabarcık Sıralama (Bubble Sort) algoritması ile sıralanması.
6	Hanoi kuleleri probleminin 25 disk için öz yinelemeli çözümü.
7	100000 elemanlı, eleman değerleri en kötü durumda olan dizinin Hızlı Sıralama (Quick Sort) algoritması ile sıralanması.
8	500000 adet farklı dizinin listeye eklenmesi; eklenen ilk ve son elemanın liste içinde aranması.
9	500000 adet farklı dizgi anahtarlarına sahip sayıların Eşleme (Map) veri yapısına eklenmesi; ilk ve son elemanın eşleme içinde aranması.
10	1000000 elemanlı, eleman değerleri en kötü durumda olan dizinin; dilin yerleşik sıralama algoritmaları kullanılarak sıralanması.
11	[0..1000000] aralığındaki sayıların dosyaya yazılması.

Tablo 1. Kullanılan Test Uygulamalarının Açıklamaları.

Test No	Python	Ruby	Java
1	1,165	4,228	0,056
2	61,183	105,512	8,354
3	6,636	16,854	0,182
4	71,014	248,661	8,288
5	13,370	83,744	0,278
6	39,193	86,773	0,369
7	1,318	4,979	0,054
8	0,964	4,402	2,191
9	1,252	6,427	3,221
10	0,498	1,197	2,189
11	2,720	3,964	0,846

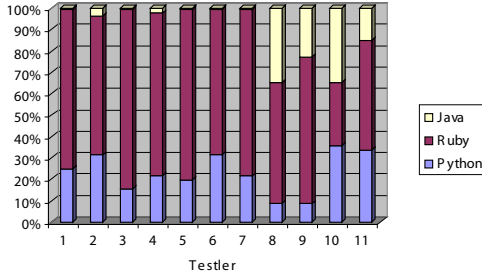
Tablo 2. Windows İşletim Sistemi Üzerinde Testlerin Çalışma Süreleri.



Şekil 1. Windows İşletim Sistemi Üzerinde Testlerin Çalışma Süreleri

Test No	Python	Ruby	Java
1	1,209	3,650	0,022
2	72,136	149,500	8,660
3	7,810	42,085	0,171
4	84,255	297,050	8,939
5	18,560	75,233	0,330
6	44,760	95,870	0,656
7	1,410	5,126	0,034
8	0,920	5,990	3,662
9	1,290	10,330	3,401
10	2,420	2,011	2,373
11	2,020	3,088	0,900

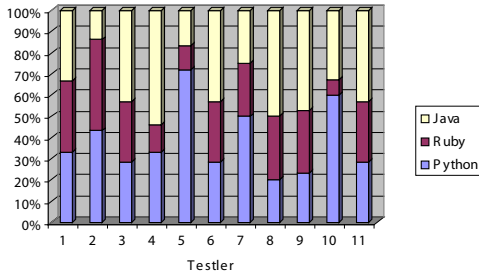
Tablo 3. Linux İşletim Sistemi Üzerinde Testlerin Çalışma Süreleri.



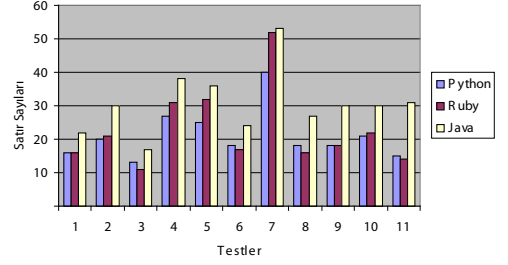
Şekil 2. Linux İşletim Sistemi Üzerinde Testlerin Çalışma Süreleri

Test No	Python	Ruby	Java
1	0,2	0,2	0,2
2	20,4	20,3	6,5
3	0,2	0,2	0,3
4	1,3	0,5	2,1
5	1,3	0,2	0,3
6	0,2	0,2	0,3
7	0,4	0,2	0,2
8	2,3	3,4	5,6
9	3,5	4,4	7,1
10	8,3	1,0	4,5
11	0,2	0,2	0,3

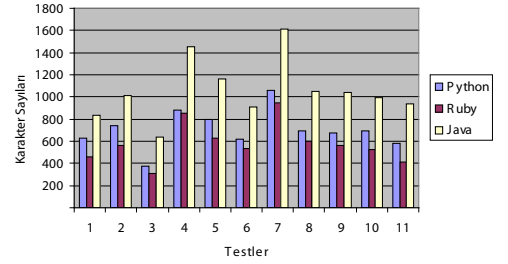
Tablo 4. Test Uygulamalarının Bellek Tüketim Yüzdeleri



Şekil 3. Linux İşletim Sistemi Üzerinde Testlerin Çalışma Süreleri



Şekil 4. Uygulamaların Satır Sayıları



Şekil 5. Uygulamalarda Kullanılan Karakter Sayısı

4. Sonuç

Yapılan testlerde, Ruby ve Python ile yazılan uygulamaların geliştirilme süreleri ve kod uzunlukları Java ile yazılan uygulamaların geliştirme süresi ve kod uzunluğunun yarısıdır. Ruby programlama dili ile yazılan uygulamalar diğer programlama dillerine göre en kısa kod uzunluğuna sahiptir. Bellek tüketimi konusunda Ruby ortalama olarak en az bellek tüketen programlama dilidir. Python programlama dili ortalama olarak en çok bellek kullanan programlama dilidir. Python programlama dili için bellek kullanımı ve ayrımı üzerine çeşitli çalışmalar yapılmaktadır[15][2]. Test uygulamalarının çalışma sürelerinde ortalama olarak Java, Python programlama diline göre 7.5 kat; Ruby programlama diline göre 21 kat daha hızlı çalışmaktadır. Java programlama dilinin ara dile derlendikten sonra yorumlanmasında bu sonuçların önemli etkisi olmasına rağmen, ortalama derleme süresi olan 1.2 sn de sonuçlara eklendiğinde sonuç üzerinde fazla bir etkisi olmamaktadır. Java programlama dili dizgi işlem-

lerinde Python programlama diline göre yavaş çalışmaktadır; Java dili aynı dizgi ifadelerinin tekrar oluşturulmadan kullanılmasını sağlayan algoritması nedeniyle bellekte bulunan tüm dizgileri kontrol eder, bu nedenle dizgi oluşturma işlemlerinde yavaştır. Performans artırımı için çeşitli çalışmalar yapılmaktadır[1][12][13][14]. Python, Ruby programlama dilleri dizi işlemlerinde listeleri kullandıkları için oldukça yavaşlardır. Dizi işlemleri için iki dil içinde C ve C++ ile geliştirilmiş dizi işlemlerinde kullanılacak modüller bulunmaktadır.

3.1 Teşekkürler

Oğuz Yarıntepe ve Pınar Yanardağ’a Python, Ruby dillerinde test uygulamalarını geliştirmemde yardımlarından dolayı teşekkür ediyorum.

4. Kaynaklar

- [1] Java Programming Language World Wide Web site, <http://java.sun.com>, Sun Microsystems.
- [2] Python Programing Language World Wide Web site, <http://www.python.org>.
- [3] Ruby Programming Language World Wide Web site, <http://www.ruby-lang.org>.
- [4] J.K.Ousterhout. “Scripting:Higher Level Programming for the 21st century”, Computer, 31(3):23-30, March 1998.
- [5] Y.F.Flu, R.J.Allan, K.C.F Maguire, “Comparing the performance of JAVA with Fortran and C for numerical computing”, Daresburg Laboratory, Germany, 2000.
- [6] Lutz Prechelt, “An empirical Comparision of Seven Programming Languages”, Computer, 0018-9162 IEEE 2000.
- [7] Benchmarks. World Wide Web site, <http://shootout.alioth.debian.org>
- [8] Python Memory Validator World Wide Web site, <http://www.softwareverify.com/pythonMemoryValidator/index.html>
- [9] Ruby Memory Validator World Wide Web site, <http://www.softwareverify.com/rubyMemoryValidator/index.html>.
- [10] Java Memory Validator World Wide Web site, <http://www.softwareverify.com/javaMemoryValidator/index.html>.
- [11] Eclipse Java Profiling Tool World Wide Web site, <http://www.eclipse.org>.
- [12] Peter Sestoft, “Java performance, reducing time and space consumption”, IT Universty of Copenhagen, Denmark, 2005.
- [13] Nick Parlante, “Java implementation and performance”, Standford Universty,2002.
- [14] J.E.Moreira, S.P.Midkiff, M.Gupta, P.V. Artigas, M.Snir, R.D.Lawrence, “Java programming for high-performance numerical computing”, IBM System Journal, vol 32, No:1, 2000.
- [15] Evan Jones, “Improving Python’s Memory Allocator”, Pycon 2005, 2005.