

AÇIKLAB GO EĞİTİMİ

PARDUS İŞLETİM SİSTEMİ ÜZERİNDE GO GELİŞTİRME

GO DİLİNİN GENEL ÖZELLİKLERİ

- Modern, makine koduna derlenebilen bir dildir
- Statically type-checked
- Bir değişkenin türü, çalışma zamanı(runtime) yerine derleme(compile) zamanında biliniyorsa, dil statik olarak türlenir
- Go, kendilerinin “*structural typing*” olarak adlandırdığı typing mantığı ile çalışır.
- Garbage Collector
- Mobil uygulama geliştirilebilir (IOS ve Android, Native)

GO DİLİNİN GENEL ÖZELLİKLERİ

- Go 1.5 sonrasına cross-compile özelliği;
- Windows executable oluşturmak için Windows'a geçmenize gerek kalmıyor :)
- *env GOOS=windows GOARCH=386 go build*

NEDEN GO

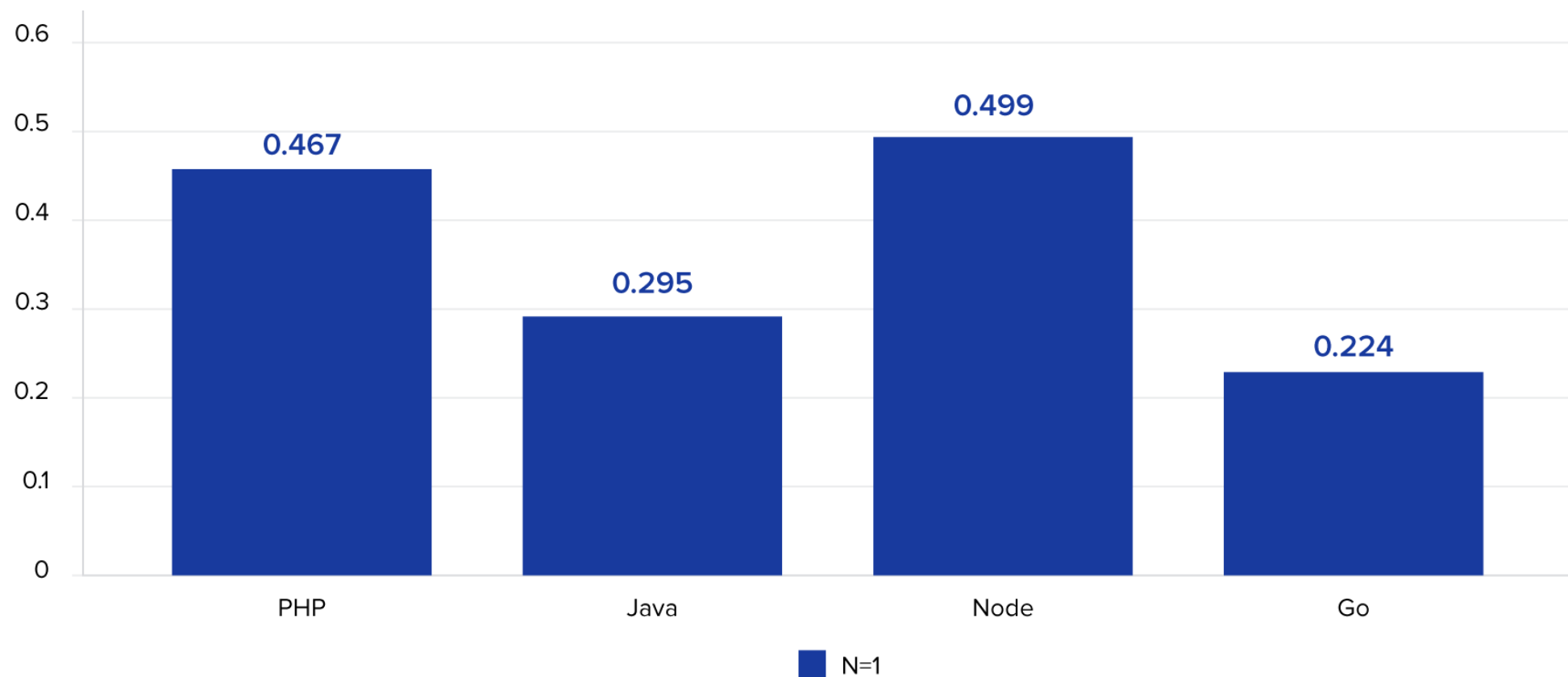
- Go, genellikle microservice mimarilerinde kullanılır
- Trendyol Zeus ekibinin bunun için hazırladığı bir makaleden alıntı olarak:
- “Grafiklerde de gördüğünüz üzere 1.2M rpm’e çıkmayı başardık. Bu yük altında pod başına maximum 87Mb bir memory allocation oldu. Routine (Golangde bir thread olarak düşünebilirsiniz) sayısı da belirli bir süre stabil gittikten sonra gittiğimiz bir API’den geç cevap almaya başladığımız anda artış gösterdi. Bunun sebebi ise; rutineler çok fazla blocklandı ve isteklere cevap verebilmek için daha fazla routine ayağa kalktı ve bir bottleneck (darboğaz) yaşanmasına neden oldu.”
- Yani microservice mimarilerinde sektörün yakın gelecekte kayacağı yer Go denilebilir

NEDEN GO

- Düşük compile süreleri ve yüksek performans

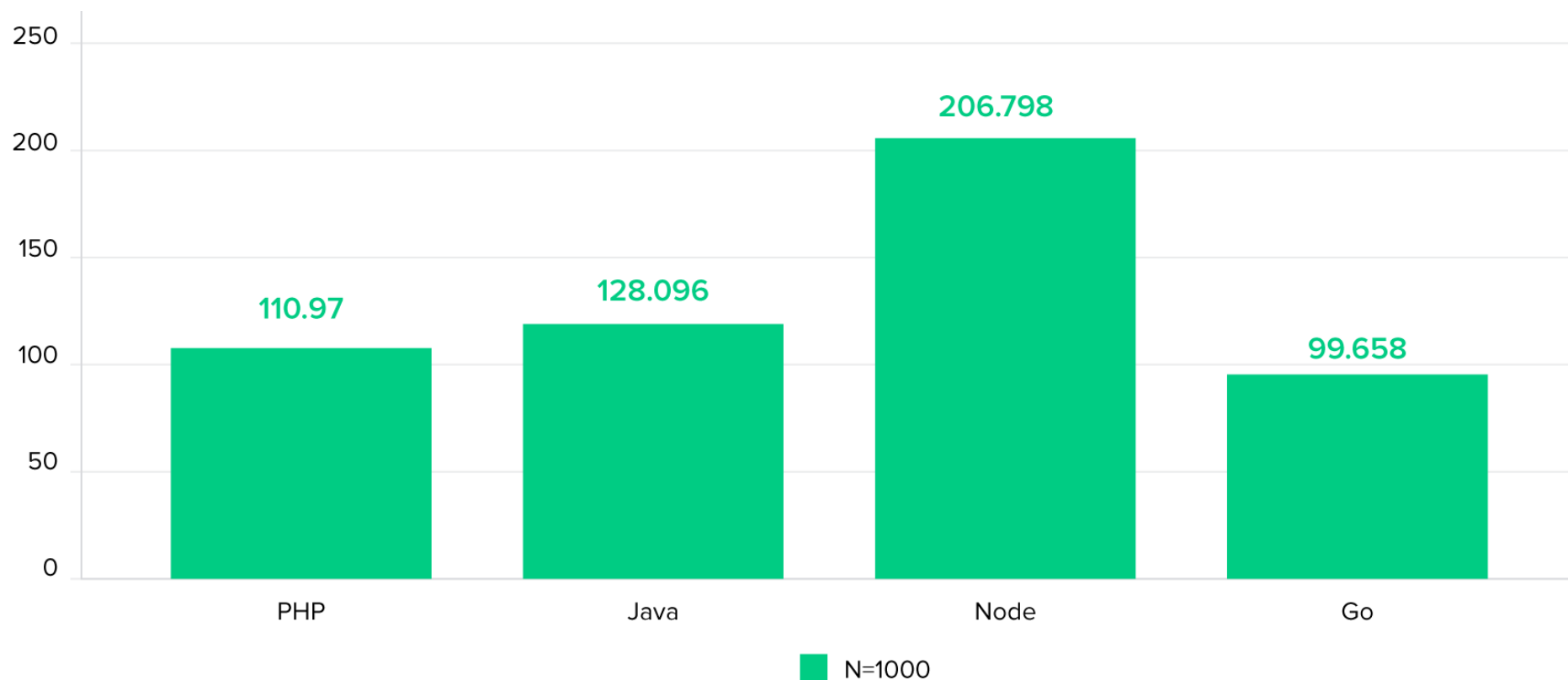
BENCHMARK

- Running 2000 iterations with 300 concurrent requests and only one hash per request (N=1)



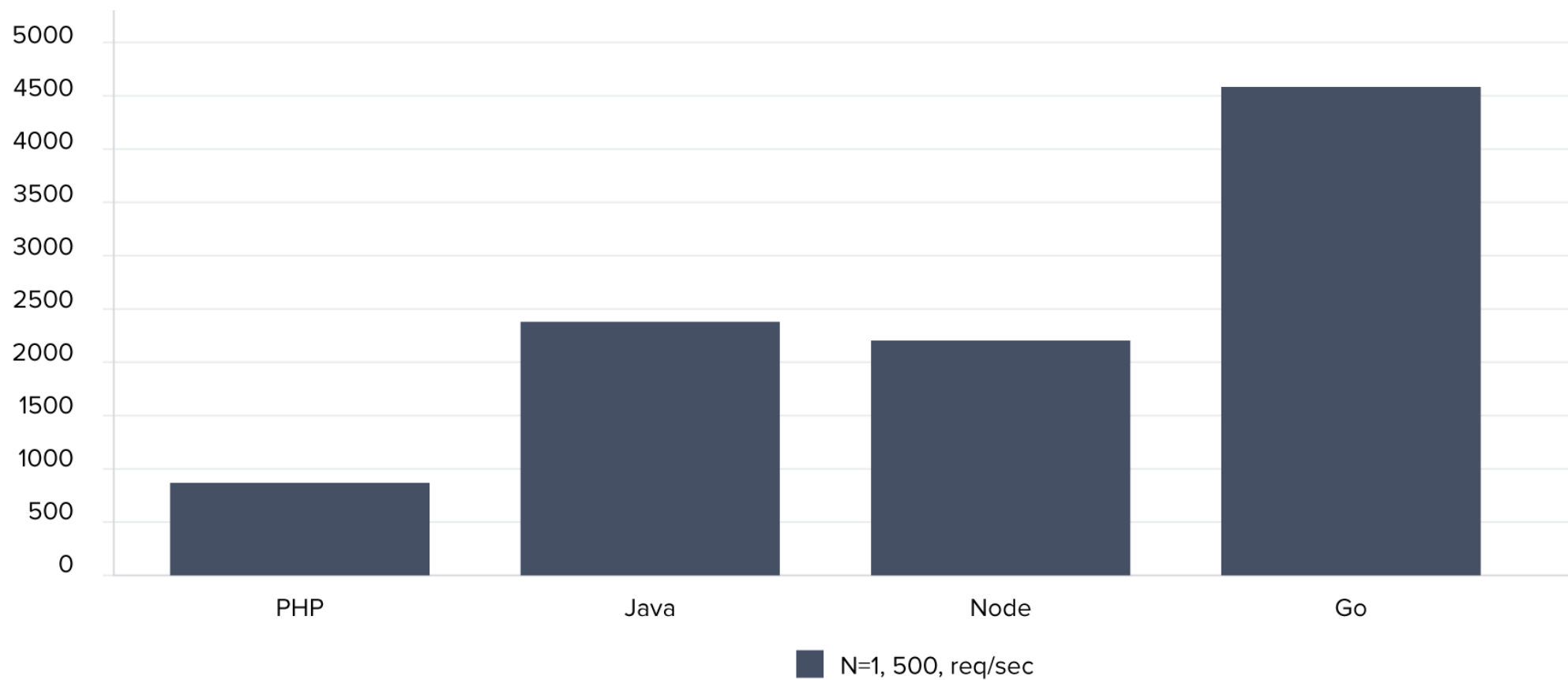
BENCHMARK

- But what happens if we increase N to 1000, still with 300 concurrent requests - the same load but 100x more hash iterations (significantly more CPU load):

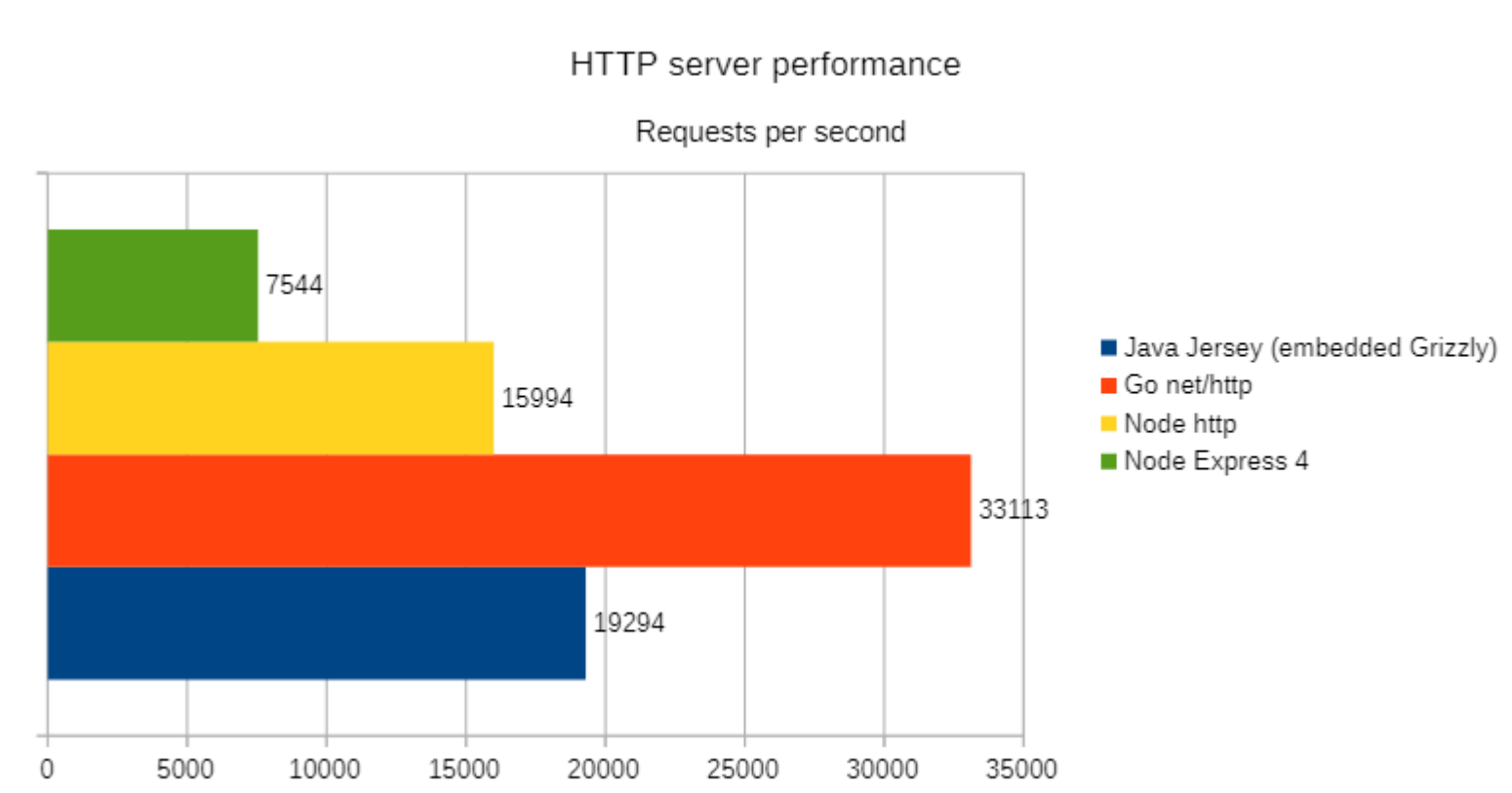


BENCHMARK

■ 5000 concurrent connections



BENCHMARK



PARA :)

- Stackoverflow survey 2020'den, en fazla kazandıran diller(Global)



PARA V2 :)

- Stackoverflow survey 2020'den, en fazla kazandıran diller(United States)



GO DİLİNİN GENEL ÖZELLİKLERİ

- go get, go install, go build, go run
- go test, go mod
- gibi komutlar ile kolaylıklar sağlar

```
package main

import "fmt"

func main() {
    fmt.Println("Hello world!")
}
```

GO DİLİNİN GENEL ÖZELLİKLERİ

- go get veya go mod tidy ile, private repo'larınızı otomatik olarak kullanabilirsiniz(github'a ssh key vermeniz gerekiyor)
- Çok detaylı ve güzel bir dokümantasyonu var

GO GELİŞTİRME ORTAMININ KURULUMU

- Basit bir golang geliştirme ortamını oluşturmak için;
- <https://golang.org/doc/install> linki takip edilerek tar dosyası indirilir
- `rm -rf /usr/local/go && tar -C /usr/local -xzf go1.16.3.linux-amd64.tar.gz`
- Komutu ile hali hazırda varsa, kurulu go'yu siler ve indirdiğimiz dosyayı, */usr/local* içine çıkarır
- Daha sonrasında *\$HOME/.profile* veya */etc/profile* içerisine `export PATH=$PATH:/usr/local/go/bin` eklenerek programın tanınması sağlanır
- Bu adımları gerçekleştirdikten sonra *go version* ile kurulumun doğruluğunu test edebiliyoruz

EDİTÖR SEÇİMİ

- Editör seçimini Visual Studio Code'dan yana seçiyorum
- Not: Eğer VSCode'u Microsoft'un oluşturduğu kurulum dosyasını kullanarak kurarsanız, Microsoft sizin datalarınızı işler
- Bkz. VSCodium

EDİTÖR

- Editörlerde sinirinizi bozacak şeylerle karşılaşabilirsiniz(importlanan paketi kullanmazsanız otomatik silme gibi)

GO MOD

- Kodunuz diğer modüllerde bulunan paketleri içe aktardığında, bu bağımlılıkları kodunuzun kendi modülü aracılığıyla yönetirsiniz. Bu modül, bu paketleri sağlayan modülleri izleyen bir go.mod dosyası tarafından tanımlanır. Bu go.mod dosyası, kaynak kodu deponuz dahil olmak üzere kodunuzda kalır.
- go mod init example

GO RUN & BUILD

- `go run main.go`
- `go run .`
- `go build main.go`
- `go build .`
- `go build -o <isim> main.go`

GO MODULU OLUŞTURMA

- `import "rsc.io/quote"`
- `go get rsc.io/quote`, veya
- `go mod tidy`

TEŞEKKÜRLER

UYGULAMAYA GEÇELİM