

Benchmarking and Profiling



Michael Van Sickle

@vansimke



Overview



Benchmark tests

- Writing
- Running

Profiling tests



Writing Benchmark Tests

main_test.go

```
package main_test
```

```
func TestFoo(t *testing.T) {
```

```
    ...
```

```
}
```

Add `_test` to filenames

Prefix tests with “Test”

Accept one parameter - `*testing.T`

Same package for whitebox tests

Add `_test` suffix to package for blackbox tests

Writing Benchmark Tests

main_test.go

```
package main_test
```

```
func BenchmarkFoo(b *testing.B) {
```

```
    ...
```

```
}
```

Add `_test` to filenames

Prefix tests with “Benchmark”

Accept one parameter - `*testing.B`

Same package for whitebox tests

Add `_test` suffix to package for blackbox tests

Key testing.B Members

b.N

b.StartTimer

b.StopTimer

b.ResetTimer

b.RunParallel



`go test`

`go test {pkg1} {pkg2} ... {pkgn}`

`go test -bench`

`go test -bench -benchtime 10s`

◀ **Run all tests in current directory**

◀ **Test specified packages**

◀ **Run all tests, including benchmark tests**

◀ **Run benchmark tests, targeting the specified time**



```
go test -benchmem
```

```
go test -trace {trace.out}
```

```
go test -{type}profile {file}
```

- ◀ Report memory allocation statistics for benchmarks
- ◀ Record execution trace to {trace.out} for analysis
- ◀ Generate profile of requested type:
 - block
 - cover
 - cpu
 - mem
 - mutex



Summary



Benchmark tests

- Writing
- Running

Profiling tests



Course Summary

**Introduction to
Testing**

**Creating and
Running Tests**

**Benchmarking and
Profiling**

