* Benchmark

Functions of the form

func BenchmarkXxx(b \*testing.B)

are considered benchmarks, and are executed by the “go test” command when >its -bench flag is provided. Benchmarks are run sequentially.

符合格式 （见上）的函数被认为是一个性能测试程序， 当带着 *-bench=“.”* （ 参数必须有！）来执行\*\*go test\*命令的时候性能测试程序就会被顺序执行。

###简单的实例代码  
*lib.go*

package lib

func Add(x int, y int) (z int) {

z = x + y

return

}

type ForTest struct {

num int ;

}

func (this \* ForTest) Loops() {

for i:=0 ; i < 10000 ; i++ {

this.num++

}

}

*lib\_test.go*

package lib

import "testing"

type AddArray struct {

result int;

add\_1 int;

add\_2 int;

}

func BenchmarkLoops(b \*testing.B) {

var test ForTest

ptr := &test

// 必须循环 b.N 次 。 这个数字 b.N 会在运行中调整，以便最终达到合适的时间消耗。方便计算出合理的数据。 （ 免得数据全部是 0 ）

for i:=0 ; i<b.N ; i++ {

ptr.Loops()

}

}

// 测试并发效率

func BenchmarkLoopsParallel(b \*testing.B) {

b.RunParallel(func(pb \*testing.PB) {

var test ForTest

ptr := &test

for pb.Next() {

ptr.Loops()

}

})

}

func TestAdd(t \* testing.T) {

var test\_data = [3] AddArray {

{ 2, 1, 1},

{ 5, 2, 3},

{ 4, 2, 2},

}

for \_ , v := range test\_data {

if v.result != Add(v.add\_1, v.add\_2) {

t.Errorf("Add( %d , %d ) != %d \n", v.add\_1 , v.add\_2 , v.result);

}

}

}

执行 *go test -bench="."* 后 结果 ：

// 表示测试全部通过

>PASS

// Benchmark 名字 - CPU 循环次数 平均每次执行时间

BenchmarkLoops-2 100000 20628 ns/op

BenchmarkLoopsParallel-2 100000 10412 ns/op

// 哪个目录下执行go test 累计耗时

ok swap/lib 2.279s