Introduction

Go is fully supported on Linux and Darwin. Any Go program that you can compile for x86/x86_64 should work on Arm. Besides Linux and Darwin, Go is also experimentally supported on FreeBSD, OpenBSD and NetBSD.

Supported architectures

Go supports the following ARM architectural families.

Architecture	Status	GOARM value	GOARCH value
ARMv4 and below	not supported	n/a	n/a
ARMv5	supported	GOARM=5	GOARCH=arm
ARMv6	supported	GOARM=6	GOARCH=arm
ARMv7	supported	GOARM=7	GOARCH=arm
ARMv8	supported	n/a	GOARCH=arm64

Starting from Go 1.1, the appropriate GOARM value will be chosen if you compile the program from source on the target machine. In cross compilation situations, it is recommended that you always set an appropriate GOARM value along with GOARCH.

Supported operating systems

- ARM on Linux. You must run an EABI kernel. These are generally known as `armel` for softfloat (compatible with ARMv5) or `armhf` for hardware floating point (ARMv6 and above).
- ARM on Darwin: ARMv7 is required.
- ARM on FreeBSD, OpenBSD, and NetBSD: ARMv6K or above is required.

Recommended Go version

Go has a mature support for ARM systems; so, just like for other architectures, use the latest stable version (eg: Go 1.15 at the time of writing).

Tips and tricks

/tmp and tmpfs

The `go` build tool uses `/tmp` when compiling and testing, this can cause heavy wear and tear if `/tmp` lives on your SD card. To minimise this effect, either `export TMPDIR` to somewhere that lives on another filesystem. Alternatively if

you have lots of physical memory you can mount a swap backed tmpfs filesystem on /tmp by adding this line to `/etc/fstab`

```
tmpfs /tmp tmpfs nodev,nosuid,mode=1777 0 0
```

Swap

Building Go from source requires at least 256mb of RAM. Running the tests requires at least 256mb of memory and at least 512mb of swap space.

Test failures due to resource starvation

The runtime tests create many native operating system threads which at the default of 8mb per thread can exhaust an ARM system with 32bit user mode address space (especially on multicore ARM systems such as the Raspberry PI 2). To prevent the runtime test from failing you may need lower the thread stack limit:

```
% ulimit -s 1024      # set the thread stack limit to 1mb
% ulimit -s           # check that it worked
1024
```

See Dave Cheney's blog post about building Go on Raspberry Pi for details.

Build failures due to lack of memory

The Go tool will try to keep all your cpu cores busy when installing packages (during make.bash), this is normally preferable on PCs where memory is abundant. However, some powerful multicore ARM machines don't have enough memory to support parallel builds utilizing all available cores, and you can work around that by using the `taskset(1)` utility to limit Go to only use one core without resorting to swaps.

```
taskset 1 ./make.bash # use 3 if you want to use two cores
```

Note: the 1 here is a bitmask for cpu affinity and it's not the number of cpu cores you're willing to use, please refer to `taskset(1)` manual for details.

Known issues

Lack of floating point hardware on ARMv5

The major issue with ARMv5 is the lack of floating point support in common ARMv5 hardware†. When compiled with the GOARM=5 environment variable, the 5l linker will insert a call to `_sfloat` before any block of floating point instructions to branch into the floating point emulator. This means that binaries produced with a Go installation that was compiled with soft float support will

work on all supported architectures, but builds compiled without soft floating point support will not work on ARMv5.

† This isn't strictly true, there exist ARMv5 implementations which have VFP1 floating point. However the compiler doesn't support VFP1 yet.

html/template and test/nilptr.go test fail on HTC Android

html/template test and test/nilptr.go is known to fail on HTC's Android kernels (ref), because the kernel will kill the application after 10 segfaults.

Potential kernel bug in 2.6.32-5-kirkwood on QNAP 219P

See Issue 5466 for details. Updating to 3.2.0-4-kirkwood solved the issue.

Success stories

ARM hardware comes in a myriad of shapes and sizes. If you've had a success story building and running Go on your Arm system, please detail your results here.

Netgear Stora

Architecture: ARMv5

Operating System: Debian Sid

The Netgear Stora is an ARMv5 (Marvell Kirkwood) platform. I flashed mine with a Debian Sid distribution and it was, until Go1, a solid platform for Go development. The main drawback is the Stora only has 128mb of ram, which is not quite enough to run ./all.bash as 5l can use more than 100mb of ram when linking some commands.

Instructions for installing Debian on your Stora can be found on the OpenStora website, http://www.openstora.com/wiki/index.php?title=How_to_install_Debian_Linux_on_NETGEAR_St

> – *dave cheney*

Qnap TS-119P II

Architecture: ARMv5

Operating System: Debian Squeeze

The Qnap TS series of NASs are excellent hackable little linux hosts. The TS-11P9 II is a 2Ghz Marvell Kirkwood ARMv5 processor with 512mb of ram and a single SATA drive bay.

The kirkwood platform is supported by the native debian installer. http://www.cyrius.com/debian/kirkwood/qnap/ts-119/install.html

Pandaboard

Architecture: ARMv7

Operating System: Ubuntu 12.04LTS (armhf)

The Pandaboard is a dual core ARMv7 development board based on the Texas Instruments OMAP4 SoC platform. I run ubuntu 12.04 LTS server on mine, which is an excellent distribution for Arm development. The Pandaboard has a gig of ram which makes it excellent for development and benchmarking.

Instructions and SD card image can be found on on the Ubuntu wiki, https://wiki.ubuntu.com/ARM/Server/Install#Installing_pre-installed_OMAP4_Precise_.2812.04.29_Server_

BeagleBone

Architecture: ARMv7 single core, Cortex-A8, 256MB RAM, 720 MHz

Operating System: Angstrom Linux

BeagleBone is similar to Beagleboard, but without the video components. Angstrom is a very small Linux distribution for ARM based systems. It is built on top of Yocto and OpenEmbedded with additional tools and recipes to make it even easier to build a distribution. You can think of Angstrom as Ubuntu and OpenEmbedded/Yocto as Debian. Angstrom is very light weight and fast compared to Ubuntu. It uses systemd instead of the sys5 scripts which help give you a very fast boot time of a few seconds.

BeagleBone is probably faster than a RasberryPI because of it's newer Cortex-A8 dual-issue superscalar architecture, but the PI has the GPU which theoretically could be used with something like OpenCL to really run circles around the BeagleBone. However, for embedded applications the BeagleBone is easier to work with because it is ready out of the box with GPIO connections.

I've cross compiled for ARM with 5g from a Mac and so far I haven't run into any problems. You can build on the BeagleBone, but cross compiling with Go is so easy that it is better to save wear and tear on the flash drive and just compile somewhere else.

Zyxel NSA 310

Architecture: ARM5 Platform: Debian Wheeze

Successfuly built default branch, going to write fan control daemon for this device in golang.

Raspberry Pi

- Building Go 1.5 on the Raspberry Pi - Dave Cheney

Architecture: ARM1176JZFS, with floating point, running at 700Mhz

Operating System: Debian Wheezy beta distribution (http://www.raspberrypi.org/archives/1435) reported as:

```
Linux raspberrypi 3.1.9+ #125 PREEMPT Sun Jun 17 16:09:36 BST
2012 armv6l GNU/Linux
```

Memory Split: the Pi shares its 256mb of memory between the CPU and the GPU. You should allocate as much memory as possible to the CPU for a successful compilation. The configuration for the memory split is stored on your SD card. This link has a script to adjust the configuration, http://sirlagz.net/?p=445.

Go version weekly.2012-03-27 +645947213cac, with timeout and GOARM 7 patches http://codereview.appspot.com/5987063/) builds with 2 test failures: encoding/gob fails with out of memory, and fmt fails the NaN test.

Successfully installed and run SVGo via go get github.com/ajstarks/svgo, tested with goplay:

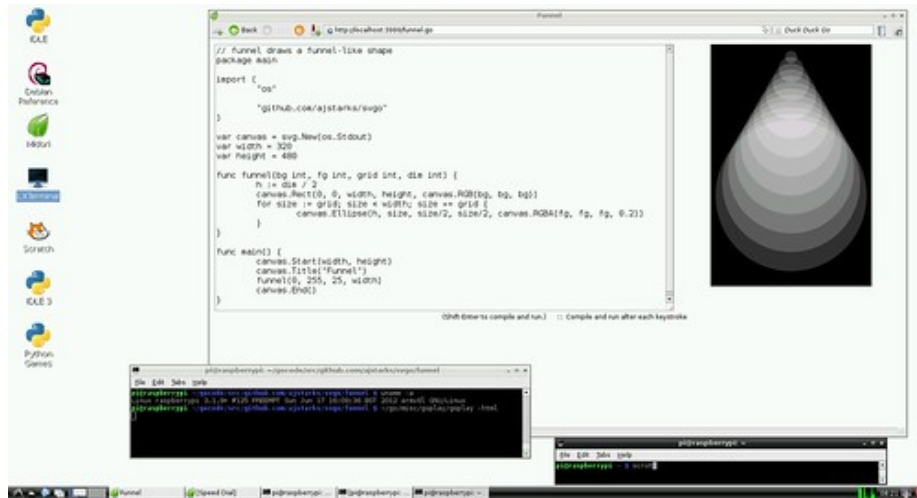Figure 1: http://farm8.staticflickr.com/7139/7451061716_fbb585c55f.jpg

Division benchmark via http://codereview.appspot.com/6258067:

```
$ cd $GOROOT/src/pkg/runtime
$ go test -test.bench=BenchmarkUint


BenchmarkUint32Div7   5000000                547 ns/op
```

```
BenchmarkUint32Div37        5000000              547 ns/op
BenchmarkUint32Div123       5000000              547 ns/op
BenchmarkUint32Div763       5000000              547 ns/op
BenchmarkUint32Div1247      5000000              547 ns/op
BenchmarkUint32Div9305      5000000              547 ns/op
BenchmarkUint32Div13307     5000000              547 ns/op
BenchmarkUint32Div52513     5000000              547 ns/op
BenchmarkUint32Div60978747     5000000              547 ns/op
BenchmarkUint32Div106956295  5000000              547 ns/op
BenchmarkUint32Mod7   5000000              547 ns/op
BenchmarkUint32Mod37        5000000              547 ns/op
BenchmarkUint32Mod123       5000000              547 ns/op
BenchmarkUint32Mod763       5000000              547 ns/op
BenchmarkUint32Mod1247      5000000              547 ns/op
BenchmarkUint32Mod9305      5000000              547 ns/op
BenchmarkUint32Mod13307     5000000              547 ns/op
BenchmarkUint32Mod52513     5000000              547 ns/op
BenchmarkUint32Mod60978747     5000000              547 ns/op
BenchmarkUint32Mod106956295  5000000              547 ns/op
```

Running the hardware floating point distribution, Raspbian "pisces" (http://www.raspbian.org/PiscesImages) and applying the patches in https://gist.github.com/3116118, here are the results of the Eleanor McHugh gospeed benchmark:

```
raspbian@pisces:~/gowork/src/github.com/feyeleanor/gospeed$ uname -a
Linux pisces 3.1.9+ #171 PREEMPT Tue Jul 17 01:08:22 BST 2012 armv6l GNU/Linux
raspbian@pisces:~/gowork/src/github.com/feyeleanor/gospeed$ go test -test.bench=".*"
PASS
BenchmarkBaselineCastInt32ToInt 100000000              13.5 ns/op
BenchmarkBaselineCastIntToInt32 100000000              13.5 ns/op
BenchmarkBaselineCastInt64ToUint64  100000000              17.8 ns/op
BenchmarkBaselineCastUint64ToInt64  100000000              17.2 ns/op
BenchmarkBaselineVariableGet        100000000              13.4 ns/op
BenchmarkBaselineVariableSet        100000000              22.4 ns/op
BenchmarkBaselineVariableGetInterface   100000000              13.5 ns/op
BenchmarkBaselineVariableSetInterface   50000000              31.3 ns/op
BenchmarkBaselineVariableIncrement  100000000              23.9 ns/op
BenchmarkBaselineVariableDecrement  100000000              23.9 ns/op
BenchmarkBaselineFieldGet   100000000              13.5 ns/op
BenchmarkBaselineFieldSet   100000000              20.9 ns/op
BenchmarkBaselineSliceGet   50000000              32.9 ns/op
BenchmarkBaselineSliceSet   50000000              34.5 ns/op
BenchmarkBaselineMapIntGet   1000000              1448 ns/op
BenchmarkBaselineMapIntSet   1000000              1968 ns/op
BenchmarkBaselineMapStringGet    1000000              1119 ns/op
BenchmarkBaselineMapStringSet    1000000              1675 ns/op
```

```
BenchmarkBaselineIf 100000000              15.0 ns/op
BenchmarkBaselineIfElse 100000000             15.0 ns/op
BenchmarkBaselineSwitchDefault  100000000          13.5 ns/op
BenchmarkBaselineSwitchOneCase  100000000          15.0 ns/op
BenchmarkBaselineSwitchTwoCases 100000000          18.0 ns/op
BenchmarkBaselineSwitchTwoCasesFallthrough  100000000            18.0 ns/op
BenchmarkBaselineForLoopIteration   50000000           42.0 ns/op
BenchmarkBaselineForReverseLoopIteration    50000000           36.0 ns/op
BenchmarkBaselineForRange   20000000          80.9 ns/op
BenchmarkBaselineForSliceLength 50000000           39.0 ns/op
BenchmarkBaselineForReverseSliceLength  50000000           36.0 ns/op
BenchmarkBaselineForLoopIteration10 20000000          119 ns/op
BenchmarkBaselineForReverseLoopIteration10  20000000            92.9 ns/op
BenchmarkBaselineForRange10 10000000          215 ns/op
BenchmarkBaselineForSliceLength10   20000000          109 ns/op
BenchmarkBaselineForReverseSliceLength10    20000000           92.9 ns/op
BenchmarkBaselineForLoopIteration100    2000000          929 ns/op
BenchmarkBaselineForReverseLoopIteration100 5000000          700 ns/op
BenchmarkBaselineForRange100    1000000         1567 ns/op
BenchmarkBaselineForSliceLength100  2000000          853 ns/op
BenchmarkBaselineForReverseSliceLength100   5000000          700 ns/op
BenchmarkBaselineForLoopIteration10000      10000       106006 ns/op
BenchmarkBaselineForReverseLoopIteration10000       50000        67480 ns/op
BenchmarkBaselineForRange10000      10000       153841 ns/op
BenchmarkBaselineForSliceLength10000        20000        85735 ns/op
BenchmarkBaselineForReverseSliceLength10000     50000        69461 ns/op
BenchmarkBaselineMakeChannelBoolUnbuffered      200000        10162 ns/op
BenchmarkBaselineMakeChannelBool1   200000        12517 ns/op
BenchmarkBaselineMakeChannelBool10      200000        12521 ns/op
BenchmarkBaselineMakeChannelStringUnbuffered        500000        10369 ns/op
BenchmarkBaselineMakeChannelString1 200000        12576 ns/op
BenchmarkBaselineMakeChannelString10        100000        22358 ns/op
BenchmarkBaselineGo     50000       367593 ns/op
BenchmarkBaselineFunctionCall   50000000           57.0 ns/op
BenchmarkBaselineFunctionCallArg    20000000           81.0 ns/op
BenchmarkBaselineFunctionCall5VarArgs       500000         6852 ns/op
BenchmarkBaselineFunctionCallInt    50000000           60.3 ns/op
BenchmarkBaselineFunctionCall5VarInts   1000000         3185 ns/op
BenchmarkBaselineFunctionCallWithDefer   1000000         2330 ns/op
BenchmarkBaselineFunctionCallPanicRecover       500000         6222 ns/op
BenchmarkBaselineMethodCallDirect   20000000           83.8 ns/op
BenchmarkBaselineMethodCallDirect1Arg   20000000          106 ns/op
BenchmarkBaselineMethodCallDirect1Int   20000000           85.2 ns/op
BenchmarkBaselineMethodCallDirect5Args  5000000          368 ns/op
BenchmarkBaselineMethodCallDirect5Ints  10000000          233 ns/op
BenchmarkBaselineMethodCallIndirect 100000000           18.0 ns/op
```

```
BenchmarkBaselineMethodCallIndirect1Arg 50000000              42.0 ns/op
BenchmarkBaselineMethodCallIndirect1Int 100000000            19.5 ns/op
BenchmarkBaselineMethodCallIndirect5Args    5000000              309 ns/op
BenchmarkBaselineMethodCallIndirect5Ints    10000000             168 ns/op
BenchmarkBaselineTypeAssertion  10000000            218 ns/op
BenchmarkBaselineTypeAssertionEmptyInterface    20000000            106 ns/op
BenchmarkBaselineTypeAssertionInterface1     5000000          576 ns/op
BenchmarkBaselineTypeAssertionInterface2     5000000          579 ns/op
BenchmarkBaselineTypeReflectPrimitiveToValue     5000000            425 ns/op
BenchmarkBaselineTypeReflectSliceToValue    1000000         3218 ns/op
BenchmarkBaselineTypeReflectStructToValue     500000          4760 ns/op
BenchmarkBaselineTypeCheck  10000000            189 ns/op
BenchmarkBaselineTypeCheckEmptyInterface    20000000            93.1 ns/op
BenchmarkBaselineTypeCheckInterface1     5000000            511 ns/op
BenchmarkBaselineTypeCheckInterface2     5000000            516 ns/op
BenchmarkBaselineTypeSwitchOneCase  10000000            262 ns/op
BenchmarkBaselineTypeSwitchBasicTypesCase    10000000            295 ns/op
BenchmarkBaselineTypeSwitchEmptyInterface    10000000            163 ns/op
BenchmarkBaselineTypeSwitchInterface1     5000000          588 ns/op
BenchmarkBaselineTypeSwitchInterface2     5000000          602 ns/op
BenchmarkBaselineNewStructureLiteral     20000000            84.0 ns/op
BenchmarkBaselineNewStructure   20000000            127 ns/op
BenchmarkBaselineNewSliceLiteral     50000000            54.2 ns/op
BenchmarkBaselineNewSlice     1000000          3124 ns/op
BenchmarkBaselineNewMapLiteralIntToInt     500000          9083 ns/op
BenchmarkBaselineNewMapLiteralIntToInterface      500000          9807 ns/op
BenchmarkBaselineNewMapLiteralStringToInt     500000          9792 ns/op
BenchmarkBaselineNewMapLiteralStringToInterface    500000           10595 ns/op
BenchmarkBaselineNewMapLiteralIntToInt2Item    200000          14265 ns/op
BenchmarkBaselineNewMapLiteralIntToInterface2Item      200000           14669 ns/op
BenchmarkBaselineNewMapLiteralStringToInt2Item     200000          14025 ns/op
BenchmarkBaselineNewMapLiteralStringToInterface2Item     200000          15086 ns/op
BenchmarkBaselineNewMapIntToInt     500000          9025 ns/op
BenchmarkBaselineNewMapIntToInterface      500000          9753 ns/op
BenchmarkBaselineNewMapStringToInt     500000          9740 ns/op
BenchmarkBaselineNewMapStringToInterface      500000           10486 ns/op
BenchmarkBaselineSliceCopy    5000000          300 ns/op
BenchmarkBaselineNewSliceAppendElement1  1000000          3318 ns/op
BenchmarkBaselineNewSliceAppendElement10     1000000          5174 ns/op
ok      github.com/feyeleanor/gospeed    417.296s
```

– anthony starks

Raspberry Pi 2

- Building Go 1.5 on the Raspberry Pi - Dave Cheney

```
go version
go version devel +07f9c25 Wed Dec 9 21:25:05 2015 +0000 linux/arm

$ go test -timeout 20m -v -bench=Benchmark -run=X
PASS
BenchmarkAppend-4                       3000000             402 ns/op
BenchmarkAppendGrowByte-4                    50        27296836 ns/op
BenchmarkAppendGrowString-4                   1      1277592542 ns/op
BenchmarkAppend1Byte-4                 20000000            75.6 ns/op
BenchmarkAppend4Bytes-4                20000000            88.2 ns/op
BenchmarkAppend7Bytes-4                20000000             103 ns/op
BenchmarkAppend8Bytes-4                20000000            89.1 ns/op
BenchmarkAppend15Bytes-4               20000000             109 ns/op
BenchmarkAppend16Bytes-4               20000000            94.5 ns/op
BenchmarkAppend32Bytes-4               20000000            91.4 ns/op
BenchmarkAppendStr1Byte-4              20000000            73.9 ns/op
BenchmarkAppendStr4Bytes-4             20000000            84.7 ns/op
BenchmarkAppendStr8Bytes-4             20000000            88.7 ns/op
BenchmarkAppendStr16Bytes-4            20000000            94.5 ns/op
BenchmarkAppendStr32Bytes-4            20000000            91.3 ns/op
BenchmarkAppendSpecialCase-4           2000000             675 ns/op
BenchmarkCopy1Byte-4                   20000000             109 ns/op          9.13 MB/s
BenchmarkCopy2Byte-4                   20000000             112 ns/op         17.77 MB/s
BenchmarkCopy4Byte-4                   10000000             120 ns/op         33.26 MB/s
BenchmarkCopy8Byte-4                   10000000             122 ns/op         65.32 MB/s
BenchmarkCopy12Byte-4                  10000000             126 ns/op         94.82 MB/s
BenchmarkCopy16Byte-4                  10000000             129 ns/op        123.60 MB/s
BenchmarkCopy32Byte-4                  10000000             126 ns/op        252.60 MB/s
BenchmarkCopy128Byte-4                 10000000             162 ns/op        786.76 MB/s
BenchmarkCopy1024Byte-4                3000000             479 ns/op        2134.17 MB/s
BenchmarkCopy1String-4                 20000000             100 ns/op          9.98 MB/s
BenchmarkCopy2String-4                 20000000             104 ns/op         19.19 MB/s
BenchmarkCopy4String-4                 20000000             111 ns/op         35.86 MB/s
BenchmarkCopy8String-4                 20000000             114 ns/op         70.02 MB/s
BenchmarkCopy12String-4                20000000             116 ns/op        103.17 MB/s
BenchmarkCopy16String-4                10000000             120 ns/op        132.69 MB/s
BenchmarkCopy32String-4                20000000             116 ns/op        273.77 MB/s
BenchmarkCopy128String-4               10000000             150 ns/op        851.31 MB/s
BenchmarkCopy1024String-4              3000000             472 ns/op        2167.03 MB/s
BenchmarkChanNonblocking-4             20000000            77.7 ns/op
BenchmarkSelectUncontended-4           2000000             780 ns/op
BenchmarkSelectSyncContended-4          100000           15094 ns/op
BenchmarkSelectAsyncContended-4         500000            2569 ns/op
BenchmarkSelectNonblock-4             10000000             226 ns/op
BenchmarkChanUncontended-4               50000           26993 ns/op
BenchmarkChanContended-4                 10000          111382 ns/op
```

```
BenchmarkChanSync-4                    300000      3994 ns/op
BenchmarkChanProdCons0-4               500000      3100 ns/op
BenchmarkChanProdCons10-4             1000000      2099 ns/op
BenchmarkChanProdCons100-4            1000000      1342 ns/op
BenchmarkChanProdConsWork0-4           500000      3071 ns/op
BenchmarkChanProdConsWork10-4          500000      2332 ns/op
BenchmarkChanProdConsWork100-4        1000000      1382 ns/op
BenchmarkSelectProdCons-4              300000      5015 ns/op
BenchmarkChanCreation-4               2000000       629 ns/op
BenchmarkChanSem-4                    1000000      1116 ns/op
BenchmarkChanPopular-4                    200   7597153 ns/op
BenchmarkCallClosure-4               30000000      43.5 ns/op
BenchmarkCallClosure1-4              30000000      49.0 ns/op
BenchmarkCallClosure2-4              5000000        323 ns/op
BenchmarkCallClosure3-4              5000000        318 ns/op
BenchmarkCallClosure4-4              5000000        324 ns/op
BenchmarkComplex128DivNormal-4       3000000        488 ns/op
BenchmarkComplex128DivNisNaN-4       5000000        375 ns/op
BenchmarkComplex128DivDisNaN-4       5000000        362 ns/op
BenchmarkComplex128DivNisInf-4       5000000        289 ns/op
BenchmarkComplex128DivDisInf-4       5000000        273 ns/op
BenchmarkSetTypePtr-4               20000000       85.1 ns/op    46.98 MB/s
BenchmarkSetTypePtr8-4             10000000        169 ns/op    189.14 MB/s
BenchmarkSetTypePtr16-4            10000000        213 ns/op    299.83 MB/s
BenchmarkSetTypePtr32-4             5000000        297 ns/op    429.62 MB/s
BenchmarkSetTypePtr64-4             3000000        462 ns/op    553.72 MB/s
BenchmarkSetTypePtr126-4            2000000        791 ns/op    636.59 MB/s
BenchmarkSetTypePtr128-4            2000000        777 ns/op    658.60 MB/s
BenchmarkSetTypePtrSlice-4           200000       6208 ns/op    659.75 MB/s
BenchmarkSetTypeNode1-4            10000000        160 ns/op     74.86 MB/s
BenchmarkSetTypeNode1Slice-4        1000000       1076 ns/op    356.60 MB/s
BenchmarkSetTypeNode8-4            10000000        221 ns/op    180.94 MB/s
BenchmarkSetTypeNode8Slice-4        1000000       2359 ns/op    542.53 MB/s
BenchmarkSetTypeNode64-4            3000000        506 ns/op    521.63 MB/s
BenchmarkSetTypeNode64Slice-4        100000      12992 ns/op    650.22 MB/s
BenchmarkSetTypeNode64Dead-4        5000000        308 ns/op    856.90 MB/s
BenchmarkSetTypeNode64DeadSlice-4     200000      11506 ns/op    734.21 MB/s
BenchmarkSetTypeNode124-4           2000000        799 ns/op    630.27 MB/s
BenchmarkSetTypeNode124Slice-4       100000      23306 ns/op    692.00 MB/s
BenchmarkSetTypeNode126-4           2000000        776 ns/op    659.33 MB/s
BenchmarkSetTypeNode126Slice-4       100000      21520 ns/op    761.31 MB/s
BenchmarkSetTypeNode128-4           2000000        850 ns/op    611.35 MB/s
BenchmarkSetTypeNode128Slice-4        50000      24122 ns/op    689.81 MB/s
BenchmarkSetTypeNode130-4           2000000        827 ns/op    638.06 MB/s
BenchmarkSetTypeNode130Slice-4        50000      24322 ns/op    694.67 MB/s
BenchmarkSetTypeNode1024-4           300000       5655 ns/op    725.66 MB/s
```

```
BenchmarkSetTypeNode1024Slice-4        10000       183602 ns/op    715.28 MB/s
BenchmarkAllocation-4                  10000       166825 ns/op
BenchmarkHash5-4                    10000000          217 ns/op     23.02 MB/s
BenchmarkHash16-4                   5000000          289 ns/op     55.18 MB/s
BenchmarkHash64-4                   2000000          770 ns/op     83.10 MB/s
BenchmarkHash1024-4                  200000         9442 ns/op    108.44 MB/s
BenchmarkHash65536-4                   2000       600452 ns/op    109.14 MB/s
BenchmarkEqEfaceConcrete-4         20000000         79.1 ns/op
BenchmarkEqIfaceConcrete-4         20000000         77.0 ns/op
BenchmarkNeEfaceConcrete-4         20000000         80.0 ns/op
BenchmarkNeIfaceConcrete-4         20000000         77.7 ns/op
BenchmarkConvT2ESmall-4             5000000          362 ns/op
BenchmarkConvT2EUintptr-4           5000000          394 ns/op
BenchmarkConvT2ELarge-4             3000000          457 ns/op
BenchmarkConvT2ISmall-4             3000000          482 ns/op
BenchmarkConvT2IUintptr-4           3000000          524 ns/op
BenchmarkConvT2ILarge-4             3000000          600 ns/op
BenchmarkConvI2E-4                 20000000         60.5 ns/op
BenchmarkConvI2I-4                  5000000          302 ns/op
BenchmarkAssertE2T-4               10000000          121 ns/op
BenchmarkAssertE2TLarge-4          10000000          131 ns/op
BenchmarkAssertE2I-4               5000000          329 ns/op
BenchmarkAssertI2T-4              10000000          125 ns/op
BenchmarkAssertI2I-4              5000000          328 ns/op
BenchmarkAssertI2E-4             20000000         84.6 ns/op
BenchmarkAssertE2E-4             50000000         32.5 ns/op
BenchmarkAssertE2T2-4            10000000          129 ns/op
BenchmarkAssertE2T2Blank-4      100000000         18.0 ns/op
BenchmarkAssertI2E2-4           20000000         91.2 ns/op
BenchmarkAssertI2E2Blank-4     100000000         16.7 ns/op
BenchmarkAssertE2E2-4           10000000          159 ns/op
BenchmarkAssertE2E2Blank-4     100000000         16.8 ns/op
BenchmarkMalloc8-4              5000000          317 ns/op
BenchmarkMalloc16-4            3000000          485 ns/op
BenchmarkMallocTypeInfo8-4     3000000          587 ns/op
BenchmarkMallocTypeInfo16-4    2000000          661 ns/op
BenchmarkMallocLargeStruct-4    500000         3205 ns/op
BenchmarkGoroutineSelect-4         100     18605318 ns/op
BenchmarkGoroutineBlocking-4       100     17222169 ns/op
BenchmarkGoroutineForRange-4       100     19092854 ns/op
BenchmarkGoroutineIdle-4           100     12554944 ns/op
BenchmarkMapPop100-4              5000       270721 ns/op
BenchmarkMapPop1000-4             300      4674884 ns/op
BenchmarkMapPop10000-4            10    110070793 ns/op
BenchmarkHashStringSpeed-4     3000000          400 ns/op
BenchmarkHashBytesSpeed-4      2000000          709 ns/op
```

```
BenchmarkHashInt32Speed-4                5000000         305 ns/op
BenchmarkHashInt64Speed-4                5000000         349 ns/op
BenchmarkHashStringArraySpeed-4          2000000         911 ns/op
BenchmarkMegMap-4                        5000000         367 ns/op
BenchmarkMegOneMap-4                     5000000         311 ns/op
BenchmarkMegEqMap-4                          100    10147332 ns/op
BenchmarkMegEmptyMap-4                   10000000        138 ns/op
BenchmarkSmallStrMap-4                    5000000         367 ns/op
BenchmarkMapStringKeysEight_16-4         5000000         395 ns/op
BenchmarkMapStringKeysEight_32-4         5000000         378 ns/op
BenchmarkMapStringKeysEight_64-4         5000000         378 ns/op
BenchmarkMapStringKeysEight_1M-4         5000000         376 ns/op
BenchmarkIntMap-4                        10000000        198 ns/op
BenchmarkRepeatedLookupStrMapKey32-4     2000000         799 ns/op
BenchmarkRepeatedLookupStrMapKey1M-4         100    10023558 ns/op
BenchmarkNewEmptyMap-4                    2000000         841 ns/op              0 B/op
BenchmarkNewSmallMap-4                    1000000        2357 ns/op              0 B/op
BenchmarkMapIter-4                        1000000        2132 ns/op
BenchmarkMapIterEmpty-4                   20000000        107 ns/op
BenchmarkSameLengthMap-4                  20000000        111 ns/op
BenchmarkBigKeyMap-4                      2000000         727 ns/op
BenchmarkBigValMap-4                      2000000         754 ns/op
BenchmarkSmallKeyMap-4                    5000000         296 ns/op
BenchmarkComplexAlgMap-4                  1000000        1786 ns/op
BenchmarkMemmove0-4                       30000000       47.7 ns/op
BenchmarkMemmove1-4                       30000000       50.2 ns/op        19.93 MB/s
BenchmarkMemmove2-4                       30000000       53.7 ns/op        37.28 MB/s
BenchmarkMemmove3-4                       30000000       56.9 ns/op        52.76 MB/s
BenchmarkMemmove4-4                       20000000       61.1 ns/op        65.42 MB/s
BenchmarkMemmove5-4                       20000000       75.8 ns/op        65.95 MB/s
BenchmarkMemmove6-4                       20000000       79.2 ns/op        75.80 MB/s
BenchmarkMemmove7-4                       20000000       82.5 ns/op        84.84 MB/s
BenchmarkMemmove8-4                       20000000       64.8 ns/op       123.37 MB/s
BenchmarkMemmove9-4                       20000000       67.1 ns/op       134.16 MB/s
BenchmarkMemmove10-4                      20000000       76.4 ns/op       130.87 MB/s
BenchmarkMemmove11-4                      20000000       81.3 ns/op       135.30 MB/s
BenchmarkMemmove12-4                      20000000       66.8 ns/op       179.68 MB/s
BenchmarkMemmove13-4                      20000000       70.6 ns/op       184.23 MB/s
BenchmarkMemmove14-4                      20000000       75.2 ns/op       186.19 MB/s
BenchmarkMemmove15-4                      20000000       79.5 ns/op       188.74 MB/s
BenchmarkMemmove16-4                      20000000       71.8 ns/op       222.73 MB/s
BenchmarkMemmove32-4                      20000000       68.3 ns/op       468.52 MB/s
BenchmarkMemmove64-4                      20000000       79.2 ns/op       808.35 MB/s
BenchmarkMemmove128-4                     20000000        101 ns/op      1256.13 MB/s
BenchmarkMemmove256-4                     10000000        145 ns/op      1755.64 MB/s
BenchmarkMemmove512-4                     5000000         244 ns/op      2095.34 MB/s
```

```
BenchmarkMemmove1024-4                  3000000       475 ns/op    2153.54 MB/s
BenchmarkMemmove2048-4                  2000000       883 ns/op    2317.95 MB/s
BenchmarkMemmove4096-4                  1000000      1809 ns/op    2262.99 MB/s
BenchmarkMemmoveUnaligned0-4           30000000      58.1 ns/op
BenchmarkMemmoveUnaligned1-4           20000000      63.3 ns/op      15.81 MB/s
BenchmarkMemmoveUnaligned2-4           20000000      66.6 ns/op      30.04 MB/s
BenchmarkMemmoveUnaligned3-4           20000000      69.9 ns/op      42.89 MB/s
BenchmarkMemmoveUnaligned4-4           20000000      95.6 ns/op      41.85 MB/s
BenchmarkMemmoveUnaligned5-4           20000000      98.6 ns/op      50.73 MB/s
BenchmarkMemmoveUnaligned6-4           20000000      99.9 ns/op      60.08 MB/s
BenchmarkMemmoveUnaligned7-4           20000000       101 ns/op      68.77 MB/s
BenchmarkMemmoveUnaligned8-4           20000000       108 ns/op      73.58 MB/s
BenchmarkMemmoveUnaligned9-4           20000000       112 ns/op      79.90 MB/s
BenchmarkMemmoveUnaligned10-4          10000000       126 ns/op      79.18 MB/s
BenchmarkMemmoveUnaligned11-4          10000000       128 ns/op      85.32 MB/s
BenchmarkMemmoveUnaligned12-4          10000000       132 ns/op      90.67 MB/s
BenchmarkMemmoveUnaligned13-4          10000000       125 ns/op     103.51 MB/s
BenchmarkMemmoveUnaligned14-4          10000000       132 ns/op     105.50 MB/s
BenchmarkMemmoveUnaligned15-4          10000000       138 ns/op     108.38 MB/s
BenchmarkMemmoveUnaligned16-4          10000000       141 ns/op     112.89 MB/s
BenchmarkMemmoveUnaligned32-4          10000000       154 ns/op     207.69 MB/s
BenchmarkMemmoveUnaligned64-4          10000000       211 ns/op     303.20 MB/s
BenchmarkMemmoveUnaligned128-4          5000000       318 ns/op     401.47 MB/s
BenchmarkMemmoveUnaligned256-4          3000000       436 ns/op     586.76 MB/s
BenchmarkMemmoveUnaligned512-4          2000000       722 ns/op     708.50 MB/s
BenchmarkMemmoveUnaligned1024-4         1000000      1296 ns/op     789.56 MB/s
BenchmarkMemmoveUnaligned2048-4          500000      2576 ns/op     794.83 MB/s
BenchmarkMemmoveUnaligned4096-4          300000      4999 ns/op     819.32 MB/s
BenchmarkMemclr5-4                      20000000      77.1 ns/op      64.82 MB/s
BenchmarkMemclr16-4                     20000000      96.3 ns/op     166.15 MB/s
BenchmarkMemclr64-4                     20000000      85.0 ns/op     753.23 MB/s
BenchmarkMemclr256-4                    10000000       125 ns/op    2040.75 MB/s
BenchmarkMemclr4096-4                    1000000      1662 ns/op    2464.13 MB/s
BenchmarkMemclr65536-4                     30000     50428 ns/op    1299.58 MB/s
BenchmarkMemclr1M-4                         2000    875472 ns/op    1197.73 MB/s
BenchmarkMemclr4M-4                          500   3529939 ns/op    1188.21 MB/s
BenchmarkMemclr8M-4                          200   7088731 ns/op    1183.37 MB/s
BenchmarkMemclr16M-4                         100  14275180 ns/op    1175.27 MB/s
BenchmarkMemclr64M-4                          20  59343321 ns/op    1130.86 MB/s
BenchmarkGoMemclr5-4                    20000000      61.4 ns/op      81.44 MB/s
BenchmarkGoMemclr16-4                   20000000      81.7 ns/op     195.78 MB/s
BenchmarkGoMemclr64-4                   20000000      69.8 ns/op     917.14 MB/s
BenchmarkGoMemclr256-4                  20000000       109 ns/op    2339.84 MB/s
BenchmarkClearFat8-4                   200000000      7.84 ns/op
BenchmarkClearFat12-4                  200000000      8.93 ns/op
BenchmarkClearFat16-4                  100000000      10.1 ns/op
```

```
BenchmarkClearFat24-4                       100000000        12.3 ns/op
BenchmarkClearFat32-4                       100000000        14.5 ns/op
BenchmarkClearFat40-4                        30000000        52.4 ns/op
BenchmarkClearFat48-4                        20000000        60.3 ns/op
BenchmarkClearFat56-4                        20000000        65.9 ns/op
BenchmarkClearFat64-4                        20000000        73.8 ns/op
BenchmarkClearFat128-4                       10000000         126 ns/op
BenchmarkClearFat256-4                       10000000         234 ns/op
BenchmarkClearFat512-4                        3000000         448 ns/op
BenchmarkClearFat1024-4                       2000000         872 ns/op
BenchmarkCopyFat8-4                         200000000        6.72 ns/op
BenchmarkCopyFat12-4                        200000000        7.80 ns/op
BenchmarkCopyFat16-4                        200000000        8.93 ns/op
BenchmarkCopyFat24-4                        100000000        11.2 ns/op
BenchmarkCopyFat32-4                        100000000        17.9 ns/op
BenchmarkCopyFat64-4                         20000000        75.0 ns/op
BenchmarkCopyFat128-4                        10000000         128 ns/op
BenchmarkCopyFat256-4                        10000000         236 ns/op
BenchmarkCopyFat512-4                         3000000         449 ns/op
BenchmarkCopyFat1024-4                        2000000         879 ns/op
BenchmarkFinalizer-4                              500     3830391 ns/op
BenchmarkFinalizerRun-4                        200000        6697 ns/op
BenchmarkSyscall-4                            5000000         296 ns/op
BenchmarkSyscallWork-4                        3000000         551 ns/op
BenchmarkSyscallExcess-4                      5000000         296 ns/op
BenchmarkSyscallExcessWork-4                  3000000         552 ns/op
BenchmarkPingPongHog-4                         100000       13315 ns/op
BenchmarkStackGrowth-4                         500000        2461 ns/op
BenchmarkStackGrowthDeep-4                       2000     1028254 ns/op
BenchmarkCreateGoroutines-4                    500000        2722 ns/op
BenchmarkCreateGoroutinesParallel-4           2000000         649 ns/op
BenchmarkCreateGoroutinesCapture-4             100000       21739 ns/op        16 B/op
BenchmarkClosureCall-4                        30000000        48.0 ns/op
BenchmarkMatmult-4                            50000000        40.3 ns/op
BenchmarkIfaceCmp100-4                          500000        2369 ns/op
BenchmarkIfaceCmpNil100-4                       500000        2475 ns/op
BenchmarkDefer-4                              1000000        1203 ns/op
BenchmarkDefer10-4                            1000000        1089 ns/op
BenchmarkDeferMany-4                          1000000        2045 ns/op
BenchmarkStackCopy-4                                1  2767373639 ns/op
BenchmarkCompareStringEqual-4                10000000         140 ns/op
BenchmarkCompareStringIdentical-4            30000000        41.4 ns/op
BenchmarkCompareStringSameLength-4           20000000        92.0 ns/op
BenchmarkCompareStringDifferentLength-4 100000000        13.4 ns/op
BenchmarkCompareStringBigUnaligned-4              100    11917034 ns/op     87.99 MB/s
BenchmarkCompareStringBig-4                       200    10163432 ns/op    103.17 MB/s
```

```
BenchmarkRuneIterate-4                   500000            3969 ns/op
BenchmarkRuneIterate2-4                  500000            3947 ns/op
BenchmarkUint32Div7-4                  20000000             102 ns/op
BenchmarkUint32Div37-4                 20000000             102 ns/op
BenchmarkUint32Div123-4                20000000             102 ns/op
BenchmarkUint32Div763-4                20000000             102 ns/op
BenchmarkUint32Div1247-4               20000000             102 ns/op
BenchmarkUint32Div9305-4               20000000             102 ns/op
BenchmarkUint32Div13307-4              20000000             102 ns/op
BenchmarkUint32Div52513-4              20000000             103 ns/op
BenchmarkUint32Div60978747-4           20000000            98.7 ns/op
BenchmarkUint32Div106956295-4          20000000           100.0 ns/op
BenchmarkUint32Mod7-4                  20000000             102 ns/op
BenchmarkUint32Mod37-4                 20000000             102 ns/op
BenchmarkUint32Mod123-4                20000000             102 ns/op
BenchmarkUint32Mod763-4                20000000             103 ns/op
BenchmarkUint32Mod1247-4               20000000             103 ns/op
BenchmarkUint32Mod9305-4               20000000             102 ns/op
BenchmarkUint32Mod13307-4              20000000             102 ns/op
BenchmarkUint32Mod52513-4              20000000             103 ns/op
BenchmarkUint32Mod60978747-4           20000000             100 ns/op
BenchmarkUint32Mod106956295-4          20000000             100 ns/op
ok      runtime 562.289s
```

Raspberry Pi Zero

Architecture: 1 GHz ARM1176JZF-S, running at 700Mhz; 512MB RAM

Operating System: Raspbian Jessie

```
$ go version
go version devel +5c24832 Sat Dec 5 00:10:40 2015 +0000 linux/arm

$ go test -timeout 20m -v -bench=Benchmark -run=X
PASS
BenchmarkAppend                         3000000             518 ns/op
BenchmarkAppendGrowByte                      20        95720661 ns/op
BenchmarkAppendGrowString                     1      2232033275 ns/op
BenchmarkAppend1Byte                   20000000            99.0 ns/op
BenchmarkAppend4Bytes                  10000000             120 ns/op
BenchmarkAppend7Bytes                  10000000             155 ns/op
BenchmarkAppend8Bytes                  10000000             132 ns/op
BenchmarkAppend15Bytes                 10000000             162 ns/op
BenchmarkAppend16Bytes                 10000000             134 ns/op
BenchmarkAppend32Bytes                 10000000             121 ns/op
BenchmarkAppendStr1Byte                20000000            97.0 ns/op
BenchmarkAppendStr4Bytes               10000000             118 ns/op
```

```
BenchmarkAppendStr8Bytes            10000000          129 ns/op
BenchmarkAppendStr16Bytes           10000000          132 ns/op
BenchmarkAppendStr32Bytes           10000000          121 ns/op
BenchmarkAppendSpecialCase           2000000          791 ns/op
BenchmarkCopy1Byte                  10000000          132 ns/op        7.54 MB/s
BenchmarkCopy2Byte                  10000000          144 ns/op       13.81 MB/s
BenchmarkCopy4Byte                  10000000          153 ns/op       26.12 MB/s
BenchmarkCopy8Byte                  10000000          164 ns/op       48.51 MB/s
BenchmarkCopy12Byte                 10000000          162 ns/op       74.03 MB/s
BenchmarkCopy16Byte                 10000000          167 ns/op       95.63 MB/s
BenchmarkCopy32Byte                 10000000          155 ns/op      205.20 MB/s
BenchmarkCopy128Byte                10000000          192 ns/op      664.06 MB/s
BenchmarkCopy1024Byte                2000000          689 ns/op     1484.28 MB/s
BenchmarkCopy1String                10000000          120 ns/op        8.27 MB/s
BenchmarkCopy2String                10000000          134 ns/op       14.85 MB/s
BenchmarkCopy4String                10000000          142 ns/op       28.01 MB/s
BenchmarkCopy8String                10000000          154 ns/op       51.81 MB/s
BenchmarkCopy12String               10000000          151 ns/op       79.01 MB/s
BenchmarkCopy16String               10000000          157 ns/op      101.82 MB/s
BenchmarkCopy32String               10000000          145 ns/op      219.52 MB/s
BenchmarkCopy128String              10000000          182 ns/op      700.88 MB/s
BenchmarkCopy1024String              2000000          976 ns/op     1048.87 MB/s
BenchmarkChanNonblocking             5000000          246 ns/op
BenchmarkSelectUncontended            500000         3610 ns/op
BenchmarkSelectSyncContended          100000        18957 ns/op
BenchmarkSelectAsyncContended         500000         3614 ns/op
BenchmarkSelectNonblock              2000000          814 ns/op
BenchmarkChanUncontended               10000       135820 ns/op
BenchmarkChanContended                 10000       133122 ns/op
BenchmarkChanSync                     200000         5870 ns/op
BenchmarkChanProdCons0                200000         5899 ns/op
BenchmarkChanProdCons10               500000         2138 ns/op
BenchmarkChanProdCons100             1000000         1557 ns/op
BenchmarkChanProdConsWork0            200000         8948 ns/op
BenchmarkChanProdConsWork10           300000         5043 ns/op
BenchmarkChanProdConsWork100          300000         4498 ns/op
BenchmarkSelectProdCons               200000        10818 ns/op
BenchmarkChanCreation                 500000         3187 ns/op
BenchmarkChanSem                     1000000         1266 ns/op
BenchmarkChanPopular                     100     14315946 ns/op
BenchmarkCallClosure                30000000         50.0 ns/op
BenchmarkCallClosure1               30000000         56.3 ns/op
BenchmarkCallClosure2                3000000          512 ns/op
BenchmarkCallClosure3                3000000          503 ns/op
BenchmarkCallClosure4                3000000          512 ns/op
BenchmarkComplex128DivNormal         2000000          735 ns/op
```

```
BenchmarkComplex128DivNisNaN             200000          6001 ns/op
BenchmarkComplex128DivDisNaN             200000          5992 ns/op
BenchmarkComplex128DivNisInf            5000000           390 ns/op
BenchmarkComplex128DivDisInf            5000000           375 ns/op
BenchmarkSetTypePtr                    10000000           119 ns/op    33.43 MB/s
BenchmarkSetTypePtr8                   10000000           227 ns/op   140.89 MB/s
BenchmarkSetTypePtr16                   5000000           285 ns/op   224.30 MB/s
BenchmarkSetTypePtr32                   5000000           388 ns/op   329.65 MB/s
BenchmarkSetTypePtr64                   3000000           591 ns/op   432.77 MB/s
BenchmarkSetTypePtr126                  1000000          1021 ns/op   493.46 MB/s
BenchmarkSetTypePtr128                  1000000          1003 ns/op   510.46 MB/s
BenchmarkSetTypePtrSlice                 200000          7427 ns/op   551.43 MB/s
BenchmarkSetTypeNode1                  10000000           217 ns/op    55.20 MB/s
BenchmarkSetTypeNode1Slice              1000000          1284 ns/op   299.03 MB/s
BenchmarkSetTypeNode8                   5000000           277 ns/op   144.17 MB/s
BenchmarkSetTypeNode8Slice               500000          2825 ns/op   453.02 MB/s
BenchmarkSetTypeNode64                  2000000           661 ns/op   398.89 MB/s
BenchmarkSetTypeNode64Slice              100000         15531 ns/op   543.92 MB/s
BenchmarkSetTypeNode64Dead              5000000           382 ns/op   689.66 MB/s
BenchmarkSetTypeNode64DeadSlice          100000         13252 ns/op   637.46 MB/s
BenchmarkSetTypeNode124                 1000000          1017 ns/op   495.27 MB/s
BenchmarkSetTypeNode124Slice              50000         28139 ns/op   573.14 MB/s
BenchmarkSetTypeNode126                 2000000          1001 ns/op   511.10 MB/s
BenchmarkSetTypeNode126Slice              50000         26535 ns/op   617.43 MB/s
BenchmarkSetTypeNode128                 1000000          1087 ns/op   478.22 MB/s
BenchmarkSetTypeNode128Slice              50000         29166 ns/op   570.52 MB/s
BenchmarkSetTypeNode130                 1000000          1067 ns/op   494.83 MB/s
BenchmarkSetTypeNode130Slice              50000         29691 ns/op   569.04 MB/s
BenchmarkSetTypeNode1024                 200000          7155 ns/op   573.51 MB/s
BenchmarkSetTypeNode1024Slice              5000        224292 ns/op   585.52 MB/s
BenchmarkAllocation                        2000        957094 ns/op
BenchmarkHash5                          5000000           349 ns/op    14.29 MB/s
BenchmarkHash16                         3000000           436 ns/op    36.66 MB/s
BenchmarkHash64                         1000000          1007 ns/op    63.49 MB/s
BenchmarkHash1024                        200000         11267 ns/op    90.88 MB/s
BenchmarkHash65536                         2000        797741 ns/op    82.15 MB/s
BenchmarkEqEfaceConcrete               20000000           111 ns/op
BenchmarkEqIfaceConcrete               20000000           106 ns/op
BenchmarkNeEfaceConcrete               20000000           111 ns/op
BenchmarkNeIfaceConcrete               20000000           106 ns/op
BenchmarkConvT2ESmall                   3000000           561 ns/op
BenchmarkConvT2EUintptr                 2000000           599 ns/op
BenchmarkConvT2ELarge                   2000000           767 ns/op
BenchmarkConvT2ISmall                   2000000           724 ns/op
BenchmarkConvT2IUintptr                 2000000           750 ns/op
BenchmarkConvT2ILarge                   2000000           907 ns/op
```

```
BenchmarkConvI2E                       20000000        76.7 ns/op
BenchmarkConvI2I                        3000000         425 ns/op
BenchmarkAssertE2T                     10000000         176 ns/op
BenchmarkAssertE2TLarge                10000000         189 ns/op
BenchmarkAssertE2I                      3000000         455 ns/op
BenchmarkAssertI2T                     10000000         176 ns/op
BenchmarkAssertI2I                      3000000         451 ns/op
BenchmarkAssertI2E                     20000000         109 ns/op
BenchmarkAssertE2E                     30000000        40.7 ns/op
BenchmarkAssertE2T2                    10000000         180 ns/op
BenchmarkAssertE2T2Blank               50000000        22.8 ns/op
BenchmarkAssertI2E2                    20000000         117 ns/op
BenchmarkAssertI2E2Blank             100000000        21.7 ns/op
BenchmarkAssertE2E2                    10000000         231 ns/op
BenchmarkAssertE2E2Blank             100000000        21.8 ns/op
BenchmarkMalloc8                       3000000         507 ns/op
BenchmarkMalloc16                      2000000         848 ns/op
BenchmarkMallocTypeInfo8               2000000        1012 ns/op
BenchmarkMallocTypeInfo16              1000000        1168 ns/op
BenchmarkMallocLargeStruct              500000        4823 ns/op
BenchmarkGoroutineSelect                    20    76172338 ns/op
BenchmarkGoroutineBlocking                  20    63469806 ns/op
BenchmarkGoroutineForRange                  20    64107650 ns/op
BenchmarkGoroutineIdle                      20    60844381 ns/op
BenchmarkMapPop100                        3000      409498 ns/op
BenchmarkMapPop1000                        200     7783817 ns/op
BenchmarkMapPop10000                        10   170736704 ns/op
BenchmarkHashStringSpeed               3000000         541 ns/op
BenchmarkHashBytesSpeed                1000000        1157 ns/op
BenchmarkHashInt32Speed                5000000         371 ns/op
BenchmarkHashInt64Speed                3000000         427 ns/op
BenchmarkHashStringArraySpeed          1000000        1417 ns/op
BenchmarkMegMap                        3000000         404 ns/op
BenchmarkMegOneMap                     5000000         346 ns/op
BenchmarkMegEqMap                          100    16663414 ns/op
BenchmarkMegEmptyMap                  10000000         160 ns/op
BenchmarkSmallStrMap                   3000000         415 ns/op
BenchmarkMapStringKeysEight_16         3000000         417 ns/op
BenchmarkMapStringKeysEight_32         3000000         420 ns/op
BenchmarkMapStringKeysEight_64         3000000         420 ns/op
BenchmarkMapStringKeysEight_1M         3000000         419 ns/op
BenchmarkIntMap                       10000000         220 ns/op
BenchmarkRepeatedLookupStrMapKey32     1000000        1015 ns/op
BenchmarkRepeatedLookupStrMapKey1M         100    15652033 ns/op
BenchmarkNewEmptyMap                   1000000        1395 ns/op        0 B/op        0 all
BenchmarkNewSmallMap                    300000        3368 ns/op        0 B/op        0 all
```

```
BenchmarkMapIter                    500000       2947 ns/op
BenchmarkMapIterEmpty             20000000        163 ns/op
BenchmarkSameLengthMap            10000000        133 ns/op
BenchmarkBigKeyMap                 2000000        899 ns/op
BenchmarkBigValMap                 2000000        922 ns/op
BenchmarkSmallKeyMap               3000000        402 ns/op
BenchmarkComplexAlgMap             1000000       2003 ns/op
BenchmarkMemmove0                 20000000       61.4 ns/op
BenchmarkMemmove1                 20000000       76.6 ns/op      13.05 MB/s
BenchmarkMemmove2                 20000000       86.3 ns/op      23.18 MB/s
BenchmarkMemmove3                 20000000       83.7 ns/op      35.83 MB/s
BenchmarkMemmove4                 20000000       88.0 ns/op      45.48 MB/s
BenchmarkMemmove5                 20000000        110 ns/op      45.37 MB/s
BenchmarkMemmove6                 20000000        115 ns/op      51.99 MB/s
BenchmarkMemmove7                 10000000        123 ns/op      56.89 MB/s
BenchmarkMemmove8                 20000000       99.5 ns/op      80.44 MB/s
BenchmarkMemmove9                 20000000        112 ns/op      79.76 MB/s
BenchmarkMemmove10                10000000        125 ns/op      79.48 MB/s
BenchmarkMemmove11                10000000        125 ns/op      87.84 MB/s
BenchmarkMemmove12                20000000       96.9 ns/op     123.87 MB/s
BenchmarkMemmove13                20000000        110 ns/op     117.83 MB/s
BenchmarkMemmove14                10000000        123 ns/op     113.65 MB/s
BenchmarkMemmove15                10000000        122 ns/op     122.30 MB/s
BenchmarkMemmove16                20000000        101 ns/op     156.89 MB/s
BenchmarkMemmove32                20000000       90.8 ns/op     352.44 MB/s
BenchmarkMemmove64                20000000        108 ns/op     588.34 MB/s
BenchmarkMemmove128               10000000        127 ns/op    1001.83 MB/s
BenchmarkMemmove256               10000000        178 ns/op    1436.19 MB/s
BenchmarkMemmove512                5000000        326 ns/op    1569.07 MB/s
BenchmarkMemmove1024               2000000        631 ns/op    1620.84 MB/s
BenchmarkMemmove2048               1000000       1251 ns/op    1636.60 MB/s
BenchmarkMemmove4096                500000       2499 ns/op    1638.97 MB/s
BenchmarkMemmoveUnaligned0        20000000       72.5 ns/op
BenchmarkMemmoveUnaligned1        20000000       87.8 ns/op      11.39 MB/s
BenchmarkMemmoveUnaligned2        20000000       99.4 ns/op      20.12 MB/s
BenchmarkMemmoveUnaligned3        20000000       96.9 ns/op      30.96 MB/s
BenchmarkMemmoveUnaligned4        10000000        135 ns/op      29.44 MB/s
BenchmarkMemmoveUnaligned5        10000000        147 ns/op      33.90 MB/s
BenchmarkMemmoveUnaligned6        10000000        146 ns/op      41.09 MB/s
BenchmarkMemmoveUnaligned7        10000000        151 ns/op      46.27 MB/s
BenchmarkMemmoveUnaligned8        10000000        156 ns/op      51.22 MB/s
BenchmarkMemmoveUnaligned9        10000000        161 ns/op      55.82 MB/s
BenchmarkMemmoveUnaligned10       10000000        166 ns/op      60.09 MB/s
BenchmarkMemmoveUnaligned11       10000000        171 ns/op      64.17 MB/s
BenchmarkMemmoveUnaligned12       10000000        177 ns/op      67.79 MB/s
BenchmarkMemmoveUnaligned13       10000000        181 ns/op      71.60 MB/s
```

```
BenchmarkMemmoveUnaligned14          10000000           189 ns/op      74.03 MB/s
BenchmarkMemmoveUnaligned15          10000000           196 ns/op      76.30 MB/s
BenchmarkMemmoveUnaligned16          10000000           200 ns/op      79.91 MB/s
BenchmarkMemmoveUnaligned32          10000000           221 ns/op     144.64 MB/s
BenchmarkMemmoveUnaligned64           5000000           290 ns/op     220.16 MB/s
BenchmarkMemmoveUnaligned128          3000000           413 ns/op     309.84 MB/s
BenchmarkMemmoveUnaligned256          2000000           766 ns/op     334.16 MB/s
BenchmarkMemmoveUnaligned512          2000000           925 ns/op     553.13 MB/s
BenchmarkMemmoveUnaligned1024          500000          2747 ns/op     372.66 MB/s
BenchmarkMemmoveUnaligned2048          500000          3180 ns/op     643.85 MB/s
BenchmarkMemmoveUnaligned4096          200000          6206 ns/op     659.99 MB/s
BenchmarkMemclr5                     20000000           105 ns/op      47.53 MB/s
BenchmarkMemclr16                    20000000           110 ns/op     144.77 MB/s
BenchmarkMemclr64                    10000000           125 ns/op     511.58 MB/s
BenchmarkMemclr256                   10000000           182 ns/op    1402.34 MB/s
BenchmarkMemclr4096                    500000          2489 ns/op    1645.22 MB/s
BenchmarkMemclr65536                    50000         39883 ns/op    1643.18 MB/s
BenchmarkMemclr1M                        2000        636812 ns/op    1646.60 MB/s
BenchmarkMemclr4M                         500       2549362 ns/op    1645.24 MB/s
BenchmarkMemclr8M                         300       5162451 ns/op    1624.93 MB/s
BenchmarkMemclr16M                        200      10300878 ns/op    1628.72 MB/s
BenchmarkMemclr64M                         30      42801198 ns/op    1567.92 MB/s
BenchmarkGoMemclr5                   20000000          73.5 ns/op      67.99 MB/s
BenchmarkGoMemclr16                  20000000          85.9 ns/op     186.24 MB/s
BenchmarkGoMemclr64                  20000000          87.8 ns/op     729.24 MB/s
BenchmarkGoMemclr256                 10000000           160 ns/op    1591.97 MB/s
BenchmarkClearFat8                  100000000          15.6 ns/op
BenchmarkClearFat12                 100000000          27.5 ns/op
BenchmarkClearFat16                 100000000          32.4 ns/op
BenchmarkClearFat24                  50000000          57.0 ns/op
BenchmarkClearFat32                  50000000          80.5 ns/op
BenchmarkClearFat40                  30000000          77.5 ns/op
BenchmarkClearFat48                  30000000           105 ns/op
BenchmarkClearFat56                  20000000          87.1 ns/op
BenchmarkClearFat64                  20000000           137 ns/op
BenchmarkClearFat128                 20000000           278 ns/op
BenchmarkClearFat256                  5000000           537 ns/op
BenchmarkClearFat512                  2000000          1293 ns/op
BenchmarkClearFat1024                 1000000          2545 ns/op
BenchmarkCopyFat8                   100000000          12.2 ns/op
BenchmarkCopyFat12                  100000000          20.3 ns/op
BenchmarkCopyFat16                  100000000          34.3 ns/op
BenchmarkCopyFat24                   50000000          27.7 ns/op
BenchmarkCopyFat32                   50000000          25.4 ns/op
BenchmarkCopyFat64                   20000000           135 ns/op
BenchmarkCopyFat128                  10000000           299 ns/op
```

```
BenchmarkCopyFat256                         2000000          571 ns/op
BenchmarkCopyFat512                         1000000         1218 ns/op
BenchmarkCopyFat1024                         500000         2665 ns/op
BenchmarkFinalizer                               50     20313669 ns/op
BenchmarkFinalizerRun                         30000        49987 ns/op
BenchmarkSyscall                            1000000         1625 ns/op
BenchmarkSyscallWork                         500000         2668 ns/op
BenchmarkSyscallExcess                      1000000         1613 ns/op
BenchmarkSyscallExcessWork                   500000         2658 ns/op
BenchmarkPingPongHog                         100000        20539 ns/op
BenchmarkStackGrowth                          50000        28473 ns/op
BenchmarkStackGrowthDeep                        500      2845263 ns/op
BenchmarkCreateGoroutines                    300000         4297 ns/op
BenchmarkCreateGoroutinesParallel            300000         4300 ns/op
BenchmarkCreateGoroutinesCapture              50000        33218 ns/op            16 B/op          1 all
BenchmarkClosureCall                       30000000         55.1 ns/op
BenchmarkMatmult                           10000000          196 ns/op
BenchmarkIfaceCmp100                         500000         2672 ns/op
BenchmarkIfaceCmpNil100                      500000         2982 ns/op
BenchmarkDefer                              1000000         1819 ns/op
BenchmarkDefer10                            1000000         1478 ns/op
BenchmarkDeferMany                           500000         3464 ns/op
BenchmarkStackCopy                                1   3714828065 ns/op
BenchmarkCompareStringEqual                10000000          160 ns/op
BenchmarkCompareStringIdentical            30000000         54.1 ns/op
BenchmarkCompareStringSameLength           20000000          111 ns/op
BenchmarkCompareStringDifferentLength     100000000         16.3 ns/op
BenchmarkCompareStringBigUnaligned              100     16695582 ns/op     62.81 MB/s
BenchmarkCompareStringBig                        100     17208318 ns/op     60.93 MB/s
BenchmarkRuneIterate                         300000         5159 ns/op
BenchmarkRuneIterate2                        300000         5158 ns/op
BenchmarkUint32Div7                        10000000          144 ns/op
BenchmarkUint32Div37                       10000000          144 ns/op
BenchmarkUint32Div123                      10000000          144 ns/op
BenchmarkUint32Div763                      10000000          148 ns/op
BenchmarkUint32Div1247                     10000000          144 ns/op
BenchmarkUint32Div9305                     10000000          144 ns/op
BenchmarkUint32Div13307                    10000000          144 ns/op
BenchmarkUint32Div52513                    10000000          144 ns/op
BenchmarkUint32Div60978747                 10000000          131 ns/op
BenchmarkUint32Div106956295                10000000          131 ns/op
BenchmarkUint32Mod7                        10000000          144 ns/op
BenchmarkUint32Mod37                       10000000          144 ns/op
BenchmarkUint32Mod123                      10000000          144 ns/op
BenchmarkUint32Mod763                      10000000          144 ns/op
BenchmarkUint32Mod1247                     10000000          144 ns/op
```

```
BenchmarkUint32Mod9305                     10000000        144 ns/op
BenchmarkUint32Mod13307                    10000000        144 ns/op
BenchmarkUint32Mod52513                    10000000        144 ns/op
BenchmarkUint32Mod60978747                 10000000        131 ns/op
BenchmarkUint32Mod106956295                10000000        131 ns/op
ok      runtime 566.969s
```

ODROID-X

Architecture: ARMv7 quad-core Cortex-A9 (Samsung Exynos 4412 1.4GHz),
1GB RAM, Mali graphics (untested).

Operating System: Archlinux ARM

Go pre-1.1 compiles out of the box. The four cores make it particularly suited
to Go multi-threaded programs. An ODROID-X2 is coming (Nov 2012) with
more RAM.

– Rémy Oudompheng

BananaPi

BananaPi has a few enhanced hardware components compare with Raspberry
Pi.

Architecture	Comments
Allwinner A20(ARM Cortex-A7 Dual-core, 1GHz, Mali400MP2 GPU)	tbc
eSATA	No worry to wear out your root SD Card
Onboard Microphone	tbc
1G Etherenet	tbc
1G RAM	tbc
Reset Switch	To reset the board ?
Power Switch	To power cycle the board ?

```
root@bpi01:/data/go13/src# cat ./buildgo.bash
#!/bin/bash
# use 1 CPU to avoid out of memory compilation issue.
time taskset 2 ./make.bash

root@bpi01:/data/go13/src# ./buildgo.bash

<snipped>

Installed Go for linux/arm in /data/go1.3
Installed commands in /data/go1.3/bin
```

```
real    9m9.222s
user    8m18.960s
sys     0m40.920s
root@bpi01:/data/go1.3/src#
```

–T.J. Yang

AppliedMicro X-Gene (ARMv8)

Architecture: ARMv8 (64-bit) 8-core, 2.4GHz, 16GB RAM

Operating Systems: Linux, Fedora 21

You will need to cross-compile a toolchain using bootstrap.bash. After you copy it to the arm64 system and set `GOROOT_BOOTSTRAP`, you can build go natively.

96Boards HiKey (ARMv8)

Architecture: ARMv8 (64-bit) 8-core, 1.2GHz, 1GB RAM

Operating System: Linux (Linaro)

Go Version: 1.5Beta1

Special Notes: Enable a swap partition (<=1GB is fine). Build process is CPU-intensive and may cause the internal 90C temperature threshold to be exceeded - keep the HiKey cool during the build.

As mentioned above, use bootstrap.sh (e.g. on Ubuntu AMD64) for ARM64, then transfer over the bootstrap tbx file, untar it, and use it as GO-ROOT_BOOTSTRAP. Check out the Go sources into a separate GOROOT, and build.

–Andrew Cencini (andrew@vapor.io)

Scaleway C1 Server

Architecture: armv7l

Operating System: Debian 8.2 (armhf)

Go Version: 1.5

The Scaleway C1 Server is a dedicated ARM server with 2GiB RAM using a SAN for storage.

I used the following guide: Building Go 1.5 on the Raspberry Pi

_ –Laurent Debacker

Jetson Nano

Jetson Nano™ from NVIDIA® is a high performance low-power single board computer built for edge Artificial Intelligence computing. It has Quad-core ARM57 based CPU clocked at 1.43 Ghz, 128-core Maxwell based GPU and 4GB of LPDDR4 memory with 25.6 GB/s bandwidth.

Architecture: ARMv8-A

Operating System: Jetson Nano Developer Kit (Ubuntu 18.04 LTS) with JetPack 4.2

Kernel: I'm using a custom compiled official kernel with patch for enabling Zswap and having the root file system in a USB SSD. The procedure for doing this is detailed in Syonyk's blog.

```
abishek@Titan:~$ uname -a
Linux Titan 4.9.140 #1 SMP PREEMPT Thu May 23 01:33:05 IST 2019 aarch64 aarch64 aarch64 GNU/
```

Go Version:

```
abishek@Titan:~$ go version
go version go1.12.5 linux/arm64
```

I used the ARMv8 version from Other Ports section of official downloads. I set the following system wide environment variables in /etc/environment.

```
GOROOT="/usr/local/go"
GOPATH="/home/abishek/Developer/go/packages"
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/gam
LANG="en_IN"
```

Benchmarks:

HW configuration : 5V 4A power via DC barrel jack. Noctua 40mm PWM fan. Samsung 840 EVO 120 GB SSD.

SW configuration : Headless. Zswap. 12 GB swap partition.

Simple benchmark.

```
abishek@Titan:~$ cd $GOROOT/src/runtime
$ go test -test.bench=BenchmarkUint
PASS
ok      runtime     329.992s
```

Extensive benchmarks.

```
$ go test -test.bench=".*" -test.timeout="60m"
goos: linux
goarch: arm64
pkg: runtime
BenchmarkMakeChan/Byte-4                      10000000              219 ns/op
```

24

```
BenchmarkMakeChan/Int-4                  5000000          298 ns/op
BenchmarkMakeChan/Ptr-4                   3000000          452 ns/op
BenchmarkMakeChan/Struct/0-4            10000000          196 ns/op
BenchmarkMakeChan/Struct/32-4            3000000          519 ns/op
BenchmarkMakeChan/Struct/40-4            3000000          592 ns/op
BenchmarkChanNonblocking-4             200000000         7.93 ns/op
BenchmarkSelectUncontended-4           10000000          124 ns/op
BenchmarkSelectSyncContended-4           200000         7358 ns/op
BenchmarkSelectAsyncContended-4         2000000          671 ns/op
BenchmarkSelectNonblock-4             100000000         22.7 ns/op
BenchmarkChanUncontended-4               200000         6956 ns/op
BenchmarkChanContended-4                  30000        51996 ns/op
BenchmarkChanSync-4                      1000000         1118 ns/op
BenchmarkChanSyncWork-4                   200000        11784 ns/op
BenchmarkChanProdCons0-4                2000000          817 ns/op
BenchmarkChanProdCons10-4               2000000          656 ns/op
BenchmarkChanProdCons100-4              3000000          561 ns/op
BenchmarkChanProdConsWork0-4            2000000          803 ns/op
BenchmarkChanProdConsWork10-4           2000000          706 ns/op
BenchmarkChanProdConsWork100-4          2000000          633 ns/op
BenchmarkSelectProdCons-4               1000000         1411 ns/op
BenchmarkChanCreation-4                10000000          206 ns/op
BenchmarkChanSem-4                      3000000          502 ns/op
BenchmarkChanPopular-4                      1000      1503920 ns/op
BenchmarkCallClosure-4                200000000         6.54 ns/op
BenchmarkCallClosure1-4               300000000         5.94 ns/op
BenchmarkCallClosure2-4                20000000         72.0 ns/op
BenchmarkCallClosure3-4                20000000         77.9 ns/op
BenchmarkCallClosure4-4                20000000         74.6 ns/op
BenchmarkComplex128DivNormal-4         20000000         72.3 ns/op
BenchmarkComplex128DivNisNaN-4         30000000         47.0 ns/op
BenchmarkComplex128DivDisNaN-4         20000000         67.6 ns/op
BenchmarkComplex128DivNisInf-4         50000000         38.7 ns/op
BenchmarkComplex128DivDisInf-4         20000000         79.3 ns/op
BenchmarkSetTypePtr-4                 200000000         8.54 ns/op   936.87 MB/s
BenchmarkSetTypePtr8-4                 50000000         37.2 ns/op  1720.14 MB/s
BenchmarkSetTypePtr16-4                30000000         45.6 ns/op  2806.20 MB/s
BenchmarkSetTypePtr32-4                20000000         60.4 ns/op  4236.17 MB/s
BenchmarkSetTypePtr64-4                20000000          110 ns/op  4615.47 MB/s
BenchmarkSetTypePtr126-4               10000000          171 ns/op  5892.51 MB/s
BenchmarkSetTypePtr128-4               10000000          169 ns/op  6030.73 MB/s
BenchmarkSetTypePtrSlice-4              1000000         1589 ns/op  5154.84 MB/s
BenchmarkSetTypeNode1-4                50000000         33.7 ns/op   712.62 MB/s
BenchmarkSetTypeNode1Slice-4           10000000          197 ns/op  3881.85 MB/s
BenchmarkSetTypeNode8-4                30000000         42.1 ns/op  1899.69 MB/s
BenchmarkSetTypeNode8Slice-4            3000000          558 ns/op  4584.96 MB/s
```

```
BenchmarkSetTypeNode64-4                 10000000        128 ns/op     4123.56 MB/s
BenchmarkSetTypeNode64Slice-4              500000       2707 ns/op     6240.73 MB/s
BenchmarkSetTypeNode64Dead-4             20000000       72.4 ns/op     7291.77 MB/s
BenchmarkSetTypeNode64DeadSlice-4          500000       3203 ns/op     5274.16 MB/s
BenchmarkSetTypeNode124-4                10000000        169 ns/op     5961.36 MB/s
BenchmarkSetTypeNode124Slice-4             300000       4505 ns/op     7159.55 MB/s
BenchmarkSetTypeNode126-4                10000000        171 ns/op     5971.04 MB/s
BenchmarkSetTypeNode126Slice-4             300000       4430 ns/op     7395.67 MB/s
BenchmarkSetTypeNode128-4                10000000        181 ns/op     5737.43 MB/s
BenchmarkSetTypeNode128Slice-4             300000       4958 ns/op     6711.13 MB/s
BenchmarkSetTypeNode130-4                10000000        183 ns/op     5764.99 MB/s
BenchmarkSetTypeNode130Slice-4             300000       5092 ns/op     6635.64 MB/s
BenchmarkSetTypeNode1024-4                1000000       1104 ns/op     7430.05 MB/s
BenchmarkSetTypeNode1024Slice-4             50000      32624 ns/op     8050.89 MB/s
BenchmarkAllocation-4                       30000      42818 ns/op
BenchmarkReadMemStats-4                     50000      26260 ns/op
BenchmarkWriteBarrier-4                   20000000       59.5 ns/op
BenchmarkBulkWriteBarrier-4              50000000       30.4 ns/op
BenchmarkScanStackNoLocals-4                   10  161375576 ns/op
BenchmarkHash5-4                         50000000       24.4 ns/op      205.24 MB/s
BenchmarkHash16-4                       100000000       22.9 ns/op      698.97 MB/s
BenchmarkHash64-4                        50000000       36.2 ns/op     1766.78 MB/s
BenchmarkHash1024-4                      10000000        144 ns/op     7062.29 MB/s
BenchmarkHash65536-4                       200000       7637 ns/op     8580.67 MB/s
BenchmarkAlignedLoad-4                  200000000       7.74 ns/op
BenchmarkUnalignedLoad-4                200000000       7.72 ns/op
BenchmarkEqEfaceConcrete-4              500000000       3.51 ns/op
BenchmarkEqIfaceConcrete-4              500000000       3.86 ns/op
BenchmarkNeEfaceConcrete-4              500000000       3.51 ns/op
BenchmarkNeIfaceConcrete-4              500000000       3.86 ns/op
BenchmarkConvT2ESmall-4                 100000000       10.9 ns/op
BenchmarkConvT2EUintptr-4               100000000       10.7 ns/op
BenchmarkConvT2ELarge-4                  20000000        119 ns/op
BenchmarkConvT2ISmall-4                 100000000       11.0 ns/op
BenchmarkConvT2IUintptr-4               100000000       10.5 ns/op
BenchmarkConvT2ILarge-4                  20000000        118 ns/op
BenchmarkConvI2E-4                      300000000       5.43 ns/op
BenchmarkConvI2I-4                       30000000       52.5 ns/op
BenchmarkAssertE2T-4                    500000000       3.79 ns/op
BenchmarkAssertE2TLarge-4               200000000       7.13 ns/op
BenchmarkAssertE2I-4                     30000000       53.3 ns/op
BenchmarkAssertI2T-4                    500000000       3.75 ns/op
BenchmarkAssertI2I-4                     30000000       54.0 ns/op
BenchmarkAssertI2E-4                    300000000       5.38 ns/op
BenchmarkAssertE2E-4                    300000000       4.91 ns/op
BenchmarkAssertE2T2-4                   300000000       4.91 ns/op
```

```
BenchmarkAssertE2T2Blank-4                     500000000          3.04 ns/op
BenchmarkAssertI2E2-4                          200000000          6.67 ns/op
BenchmarkAssertI2E2Blank-4                     500000000          3.04 ns/op
BenchmarkAssertE2E2-4                          200000000          6.32 ns/op
BenchmarkAssertE2E2Blank-4                     500000000          3.04 ns/op
BenchmarkConvT2Ezero/zero/16-4                 100000000          10.9 ns/op
BenchmarkConvT2Ezero/zero/32-4                 100000000          10.5 ns/op
BenchmarkConvT2Ezero/zero/64-4                 100000000          10.5 ns/op
BenchmarkConvT2Ezero/zero/str-4                100000000          11.9 ns/op
BenchmarkConvT2Ezero/zero/slice-4              100000000          12.8 ns/op
BenchmarkConvT2Ezero/zero/big-4                  2000000           715 ns/op
BenchmarkConvT2Ezero/nonzero/16-4               30000000          52.6 ns/op
BenchmarkConvT2Ezero/nonzero/32-4               20000000          61.4 ns/op
BenchmarkConvT2Ezero/nonzero/64-4               20000000          73.1 ns/op
BenchmarkConvT2Ezero/nonzero/str-4              10000000           128 ns/op
BenchmarkConvT2Ezero/nonzero/slice-4            10000000           160 ns/op
BenchmarkConvT2Ezero/nonzero/big-4               2000000           722 ns/op
BenchmarkMalloc8-4                              20000000          67.4 ns/op
BenchmarkMalloc16-4                             20000000           102 ns/op
BenchmarkMallocTypeInfo8-4                      20000000          99.4 ns/op
BenchmarkMallocTypeInfo16-4                     20000000           119 ns/op
BenchmarkMallocLargeStruct-4                     1000000          1034 ns/op
BenchmarkGoroutineSelect-4                           200       5243015 ns/op
BenchmarkGoroutineBlocking-4                         200       6219647 ns/op
BenchmarkGoroutineForRange-4                         200       6146714 ns/op
BenchmarkGoroutineIdle-4                             500       2738542 ns/op
BenchmarkHashStringSpeed-4                      20000000          75.0 ns/op
BenchmarkHashBytesSpeed-4                       20000000           123 ns/op
BenchmarkHashInt32Speed-4                       30000000          64.2 ns/op
BenchmarkHashInt64Speed-4                       20000000          56.2 ns/op
BenchmarkHashStringArraySpeed-4                 10000000           175 ns/op
BenchmarkMegMap-4                               30000000          51.4 ns/op
BenchmarkMegOneMap-4                           100000000          21.8 ns/op
BenchmarkMegEqMap-4                                 5000        378562 ns/op
BenchmarkMegEmptyMap-4                         200000000          9.83 ns/op
BenchmarkSmallStrMap-4                          30000000          53.3 ns/op
BenchmarkMapStringKeysEight_16-4                30000000          48.8 ns/op
BenchmarkMapStringKeysEight_32-4                30000000          47.0 ns/op
BenchmarkMapStringKeysEight_64-4                30000000          47.0 ns/op
BenchmarkMapStringKeysEight_1M-4                30000000          47.3 ns/op
BenchmarkIntMap-4                               30000000          47.9 ns/op
BenchmarkMapFirst/1-4                          100000000          15.4 ns/op
BenchmarkMapFirst/2-4                          100000000          15.6 ns/op
BenchmarkMapFirst/3-4                          100000000          15.6 ns/op
BenchmarkMapFirst/4-4                          100000000          15.6 ns/op
BenchmarkMapFirst/5-4                          100000000          15.6 ns/op
```

```
BenchmarkMapFirst/6-4                    100000000         15.4 ns/op
BenchmarkMapFirst/7-4                    100000000         15.6 ns/op
BenchmarkMapFirst/8-4                    100000000         15.4 ns/op
BenchmarkMapFirst/9-4                     50000000         35.2 ns/op
BenchmarkMapFirst/10-4                    50000000         35.4 ns/op
BenchmarkMapFirst/11-4                    50000000         35.1 ns/op
BenchmarkMapFirst/12-4                    50000000         35.1 ns/op
BenchmarkMapFirst/13-4                    50000000         35.1 ns/op
BenchmarkMapFirst/14-4                    50000000         35.1 ns/op
BenchmarkMapFirst/15-4                    50000000         35.1 ns/op
BenchmarkMapFirst/16-4                    50000000         35.1 ns/op
BenchmarkMapMid/1-4                      100000000         16.1 ns/op
BenchmarkMapMid/2-4                      100000000         19.0 ns/op
BenchmarkMapMid/3-4                      100000000         19.1 ns/op
BenchmarkMapMid/4-4                      100000000         21.3 ns/op
BenchmarkMapMid/5-4                      100000000         21.1 ns/op
BenchmarkMapMid/6-4                      100000000         23.2 ns/op
BenchmarkMapMid/7-4                      100000000         23.2 ns/op
BenchmarkMapMid/8-4                       50000000         25.6 ns/op
BenchmarkMapMid/9-4                       30000000         40.0 ns/op
BenchmarkMapMid/10-4                      30000000         55.1 ns/op
BenchmarkMapMid/11-4                      30000000         46.0 ns/op
BenchmarkMapMid/12-4                      30000000         46.0 ns/op
BenchmarkMapMid/13-4                      30000000         46.0 ns/op
BenchmarkMapMid/14-4                      50000000         46.0 ns/op
BenchmarkMapMid/15-4                      30000000         40.0 ns/op
BenchmarkMapMid/16-4                      50000000         46.4 ns/op
BenchmarkMapLast/1-4                     100000000         16.1 ns/op
BenchmarkMapLast/2-4                     100000000         19.0 ns/op
BenchmarkMapLast/3-4                     100000000         21.1 ns/op
BenchmarkMapLast/4-4                       50000000         23.2 ns/op
BenchmarkMapLast/5-4                       50000000         25.3 ns/op
BenchmarkMapLast/6-4                       50000000         27.4 ns/op
BenchmarkMapLast/7-4                       50000000         29.6 ns/op
BenchmarkMapLast/8-4                       30000000         47.4 ns/op
BenchmarkMapLast/9-4                       30000000         52.1 ns/op
BenchmarkMapLast/10-4                      30000000         56.8 ns/op
BenchmarkMapLast/11-4                      30000000         56.9 ns/op
BenchmarkMapLast/12-4                      30000000         55.6 ns/op
BenchmarkMapLast/13-4                      20000000         55.1 ns/op
BenchmarkMapLast/14-4                      20000000         51.9 ns/op
BenchmarkMapLast/15-4                      30000000         48.4 ns/op
BenchmarkMapLast/16-4                      30000000         48.4 ns/op
BenchmarkMapCycle-4                        20000000         80.9 ns/op
BenchmarkRepeatedLookupStrMapKey32-4       20000000         76.9 ns/op
BenchmarkRepeatedLookupStrMapKey1M-4           5000     241809 ns/op
```

```
BenchmarkMakeMap/[Byte]Byte-4                    5000000        378 ns/op
BenchmarkMakeMap/[Int]Int-4                      3000000        586 ns/op
BenchmarkNewEmptyMap-4                          50000000       25.3 ns/op           0 B/op
BenchmarkNewSmallMap-4                          10000000        152 ns/op           0 B/op
BenchmarkMapIter-4                               5000000        323 ns/op
BenchmarkMapIterEmpty-4                        100000000       22.5 ns/op
BenchmarkSameLengthMap-4                       100000000       18.3 ns/op
BenchmarkBigKeyMap-4                            20000000       95.4 ns/op
BenchmarkBigValMap-4                            20000000       95.5 ns/op
BenchmarkSmallKeyMap-4                          20000000       66.7 ns/op
BenchmarkMapPopulate/1-4                        20000000       83.6 ns/op           0 B/op
BenchmarkMapPopulate/10-4                        1000000       1953 ns/op         179 B/op
BenchmarkMapPopulate/100-4                         50000      25432 ns/op        3252 B/op
BenchmarkMapPopulate/1000-4                         5000     279914 ns/op       49383 B/op
BenchmarkMapPopulate/10000-4                         500    2723038 ns/op      412217 B/op
BenchmarkMapPopulate/100000-4                         50   28713410 ns/op     3588881 B/op
BenchmarkComplexAlgMap-4                        10000000        214 ns/op
BenchmarkGoMapClear/Reflexive/1-4              20000000        112 ns/op
BenchmarkGoMapClear/Reflexive/10-4            10000000        123 ns/op
BenchmarkGoMapClear/Reflexive/100-4            5000000        304 ns/op
BenchmarkGoMapClear/Reflexive/1000-4            300000       3735 ns/op
BenchmarkGoMapClear/Reflexive/10000-4            50000      34074 ns/op
BenchmarkGoMapClear/NonReflexive/1-4           5000000        379 ns/op
BenchmarkGoMapClear/NonReflexive/10-4          3000000        480 ns/op
BenchmarkGoMapClear/NonReflexive/100-4         1000000       1183 ns/op
BenchmarkGoMapClear/NonReflexive/1000-4         100000      13549 ns/op
BenchmarkGoMapClear/NonReflexive/10000-4         10000     117289 ns/op
BenchmarkMapStringConversion/32/simple-4      30000000       41.9 ns/op           0 B/op
BenchmarkMapStringConversion/32/struct-4      30000000       41.9 ns/op           0 B/op
BenchmarkMapStringConversion/32/array-4       30000000       41.8 ns/op           0 B/op
BenchmarkMapStringConversion/64/simple-4      30000000       46.3 ns/op           0 B/op
BenchmarkMapStringConversion/64/struct-4      30000000       46.3 ns/op           0 B/op
BenchmarkMapStringConversion/64/array-4       30000000       46.3 ns/op           0 B/op
BenchmarkMapPop100-4                              50000      36575 ns/op
BenchmarkMapPop1000-4                              3000     511995 ns/op
BenchmarkMapPop10000-4                              100   12112237 ns/op
BenchmarkMapAssign/Int32/256-4                 20000000       66.0 ns/op
BenchmarkMapAssign/Int32/65536-4              20000000       90.2 ns/op
BenchmarkMapAssign/Int64/256-4                 20000000       66.0 ns/op
BenchmarkMapAssign/Int64/65536-4              10000000        115 ns/op
BenchmarkMapAssign/Str/256-4                   20000000       79.9 ns/op
BenchmarkMapAssign/Str/65536-4                10000000        186 ns/op
BenchmarkMapOperatorAssign/Int32/256-4        20000000       66.2 ns/op
BenchmarkMapOperatorAssign/Int32/65536-4      20000000       95.3 ns/op
BenchmarkMapOperatorAssign/Int64/256-4        20000000       66.7 ns/op
BenchmarkMapOperatorAssign/Int64/65536-4      10000000        138 ns/op
```

```
BenchmarkMapOperatorAssign/Str/256-4          1000000       5035 ns/op
BenchmarkMapOperatorAssign/Str/65536-4        3000000        682 ns/op
BenchmarkMapAppendAssign/Int32/256-4         20000000       96.4 ns/op         47 B/op
BenchmarkMapAppendAssign/Int32/65536-4        5000000        275 ns/op         28 B/op
BenchmarkMapAppendAssign/Int64/256-4         20000000       95.7 ns/op         47 B/op
BenchmarkMapAppendAssign/Int64/65536-4        5000000        293 ns/op         28 B/op
BenchmarkMapAppendAssign/Str/256-4           10000000        318 ns/op         93 B/op
BenchmarkMapAppendAssign/Str/65536-4          3000000        465 ns/op         48 B/op
BenchmarkMapDelete/Int32/100-4               20000000        104 ns/op
BenchmarkMapDelete/Int32/1000-4              20000000       77.8 ns/op
BenchmarkMapDelete/Int32/10000-4             20000000        103 ns/op
BenchmarkMapDelete/Int64/100-4               20000000        103 ns/op
BenchmarkMapDelete/Int64/1000-4              20000000       80.3 ns/op
BenchmarkMapDelete/Int64/10000-4             20000000        104 ns/op
BenchmarkMapDelete/Str/100-4                 10000000        124 ns/op
BenchmarkMapDelete/Str/1000-4                20000000        117 ns/op
BenchmarkMapDelete/Str/10000-4               10000000        126 ns/op
BenchmarkMemmove/0-4                         300000000       5.58 ns/op
BenchmarkMemmove/1-4                         100000000       13.5 ns/op      74.30 MB/s
BenchmarkMemmove/2-4                         100000000       13.5 ns/op     148.40 MB/s
BenchmarkMemmove/3-4                         100000000       13.3 ns/op     225.03 MB/s
BenchmarkMemmove/4-4                         100000000       13.3 ns/op     299.99 MB/s
BenchmarkMemmove/5-4                         100000000       13.3 ns/op     374.96 MB/s
BenchmarkMemmove/6-4                         100000000       13.5 ns/op     444.23 MB/s
BenchmarkMemmove/7-4                         100000000       13.4 ns/op     522.62 MB/s
BenchmarkMemmove/8-4                         100000000       13.3 ns/op     600.00 MB/s
BenchmarkMemmove/9-4                         100000000       13.4 ns/op     673.43 MB/s
BenchmarkMemmove/10-4                        100000000       13.3 ns/op     750.14 MB/s
BenchmarkMemmove/11-4                        100000000       13.3 ns/op     824.91 MB/s
BenchmarkMemmove/12-4                        100000000       13.3 ns/op     899.90 MB/s
BenchmarkMemmove/13-4                        100000000       13.5 ns/op     962.98 MB/s
BenchmarkMemmove/14-4                        100000000       13.4 ns/op    1046.19 MB/s
BenchmarkMemmove/15-4                        100000000       13.3 ns/op    1124.88 MB/s
BenchmarkMemmove/16-4                        100000000       13.5 ns/op    1188.13 MB/s
BenchmarkMemmove/32-4                        100000000       13.6 ns/op    2356.67 MB/s
BenchmarkMemmove/64-4                        100000000       16.3 ns/op    3924.55 MB/s
BenchmarkMemmove/128-4                        50000000       23.9 ns/op    5354.07 MB/s
BenchmarkMemmove/256-4                        50000000       35.2 ns/op    7280.74 MB/s
BenchmarkMemmove/512-4                        20000000       68.1 ns/op    7522.83 MB/s
BenchmarkMemmove/1024-4                       20000000        113 ns/op    9060.92 MB/s
BenchmarkMemmove/2048-4                       10000000        203 ns/op   10086.84 MB/s
BenchmarkMemmove/4096-4                        5000000        383 ns/op   10668.94 MB/s
BenchmarkMemmoveUnalignedDst/0-4             100000000       10.7 ns/op
BenchmarkMemmoveUnalignedDst/1-4             100000000       15.4 ns/op      64.75 MB/s
BenchmarkMemmoveUnalignedDst/2-4             100000000       14.0 ns/op     142.50 MB/s
BenchmarkMemmoveUnalignedDst/3-4             100000000       14.0 ns/op     213.78 MB/s
```

```
BenchmarkMemmoveUnalignedDst/4-4              100000000      14.5 ns/op    275.50 MB/s
BenchmarkMemmoveUnalignedDst/5-4              100000000      14.9 ns/op    335.56 MB/s
BenchmarkMemmoveUnalignedDst/6-4              100000000      14.7 ns/op    407.71 MB/s
BenchmarkMemmoveUnalignedDst/7-4              100000000      14.9 ns/op    468.79 MB/s
BenchmarkMemmoveUnalignedDst/8-4              100000000      14.5 ns/op    550.56 MB/s
BenchmarkMemmoveUnalignedDst/9-4              100000000      14.7 ns/op    611.60 MB/s
BenchmarkMemmoveUnalignedDst/10-4             100000000      15.0 ns/op    668.48 MB/s
BenchmarkMemmoveUnalignedDst/11-4             100000000      14.8 ns/op    744.55 MB/s
BenchmarkMemmoveUnalignedDst/12-4             100000000      15.1 ns/op    792.42 MB/s
BenchmarkMemmoveUnalignedDst/13-4             100000000      15.0 ns/op    866.28 MB/s
BenchmarkMemmoveUnalignedDst/14-4             100000000      14.9 ns/op    942.03 MB/s
BenchmarkMemmoveUnalignedDst/15-4             100000000      15.0 ns/op   1000.58 MB/s
BenchmarkMemmoveUnalignedDst/16-4             100000000      14.5 ns/op   1103.84 MB/s
BenchmarkMemmoveUnalignedDst/32-4             100000000      16.7 ns/op   1921.87 MB/s
BenchmarkMemmoveUnalignedDst/64-4              50000000      32.7 ns/op   1954.46 MB/s
BenchmarkMemmoveUnalignedDst/128-4             50000000      28.1 ns/op   4560.37 MB/s
BenchmarkMemmoveUnalignedDst/256-4             30000000      47.7 ns/op   5364.41 MB/s
BenchmarkMemmoveUnalignedDst/512-4             20000000      84.9 ns/op   6029.49 MB/s
BenchmarkMemmoveUnalignedDst/1024-4            10000000       156 ns/op   6552.17 MB/s
BenchmarkMemmoveUnalignedDst/2048-4             5000000       290 ns/op   7059.63 MB/s
BenchmarkMemmoveUnalignedDst/4096-4             3000000       561 ns/op   7296.37 MB/s
BenchmarkMemmoveUnalignedSrc/0-4              100000000      10.5 ns/op
BenchmarkMemmoveUnalignedSrc/1-4              100000000      15.6 ns/op     64.12 MB/s
BenchmarkMemmoveUnalignedSrc/2-4              100000000      14.0 ns/op    142.49 MB/s
BenchmarkMemmoveUnalignedSrc/3-4              100000000      14.2 ns/op    211.08 MB/s
BenchmarkMemmoveUnalignedSrc/4-4              100000000      13.9 ns/op    287.49 MB/s
BenchmarkMemmoveUnalignedSrc/5-4              100000000      13.6 ns/op    367.36 MB/s
BenchmarkMemmoveUnalignedSrc/6-4              100000000      13.9 ns/op    432.38 MB/s
BenchmarkMemmoveUnalignedSrc/7-4              100000000      13.6 ns/op    514.32 MB/s
BenchmarkMemmoveUnalignedSrc/8-4              100000000      13.8 ns/op    579.18 MB/s
BenchmarkMemmoveUnalignedSrc/9-4              100000000      14.0 ns/op    643.05 MB/s
BenchmarkMemmoveUnalignedSrc/10-4             100000000      14.0 ns/op    715.96 MB/s
BenchmarkMemmoveUnalignedSrc/11-4             100000000      13.8 ns/op    795.76 MB/s
BenchmarkMemmoveUnalignedSrc/12-4             100000000      14.0 ns/op    855.35 MB/s
BenchmarkMemmoveUnalignedSrc/13-4             100000000      14.0 ns/op    930.87 MB/s
BenchmarkMemmoveUnalignedSrc/14-4             100000000      14.0 ns/op    998.89 MB/s
BenchmarkMemmoveUnalignedSrc/15-4             100000000      14.0 ns/op   1072.50 MB/s
BenchmarkMemmoveUnalignedSrc/16-4             100000000      14.0 ns/op   1141.96 MB/s
BenchmarkMemmoveUnalignedSrc/32-4             100000000      15.4 ns/op   2072.73 MB/s
BenchmarkMemmoveUnalignedSrc/64-4             100000000      18.5 ns/op   3450.95 MB/s
BenchmarkMemmoveUnalignedSrc/128-4             50000000      25.6 ns/op   4996.57 MB/s
BenchmarkMemmoveUnalignedSrc/256-4             50000000      37.9 ns/op   6754.79 MB/s
BenchmarkMemmoveUnalignedSrc/512-4             20000000      74.0 ns/op   6919.73 MB/s
BenchmarkMemmoveUnalignedSrc/1024-4            20000000       119 ns/op   8554.12 MB/s
BenchmarkMemmoveUnalignedSrc/2048-4            10000000       206 ns/op   9940.15 MB/s
BenchmarkMemmoveUnalignedSrc/4096-4             5000000       385 ns/op  10629.02 MB/s
```

```
BenchmarkMemclr/5-4                  100000000          14.7 ns/op   339.27 MB/s
BenchmarkMemclr/16-4                 100000000          14.9 ns/op  1075.74 MB/s
BenchmarkMemclr/64-4                 100000000          20.0 ns/op  3192.24 MB/s
BenchmarkMemclr/256-4                 50000000          34.7 ns/op  7373.17 MB/s
BenchmarkMemclr/4096-4                 5000000           288 ns/op  14199.71 MB/s
BenchmarkMemclr/65536-4                 200000          5920 ns/op  11069.74 MB/s
BenchmarkMemclr/1M-4                      10000        119686 ns/op  8761.04 MB/s
BenchmarkMemclr/4M-4                       3000        492651 ns/op  8513.73 MB/s
BenchmarkMemclr/8M-4                       2000        978378 ns/op  8573.99 MB/s
BenchmarkMemclr/16M-4                      1000       1960626 ns/op  8557.07 MB/s
BenchmarkMemclr/64M-4                       200       7860433 ns/op  8537.55 MB/s
BenchmarkGoMemclr/5-4                200000000          8.66 ns/op   577.65 MB/s
BenchmarkGoMemclr/16-4               200000000          7.03 ns/op  2277.18 MB/s
BenchmarkGoMemclr/64-4               100000000          13.3 ns/op  4810.16 MB/s
BenchmarkGoMemclr/256-4               50000000          31.5 ns/op  8122.27 MB/s
BenchmarkClearFat8-4                2000000000          1.40 ns/op
BenchmarkClearFat12-4               2000000000          1.40 ns/op
BenchmarkClearFat16-4               2000000000          1.40 ns/op
BenchmarkClearFat24-4               2000000000          1.40 ns/op
BenchmarkClearFat32-4               2000000000          1.42 ns/op
BenchmarkClearFat40-4               2000000000          1.40 ns/op
BenchmarkClearFat48-4               2000000000          1.40 ns/op
BenchmarkClearFat56-4               2000000000          1.40 ns/op
BenchmarkClearFat64-4               2000000000          1.40 ns/op
BenchmarkClearFat128-4              2000000000          1.40 ns/op
BenchmarkClearFat256-4              2000000000          1.40 ns/op
BenchmarkClearFat512-4              2000000000          1.40 ns/op
BenchmarkClearFat1024-4             2000000000          1.42 ns/op
BenchmarkCopyFat8-4                 2000000000          1.40 ns/op
BenchmarkCopyFat12-4                2000000000          1.40 ns/op
BenchmarkCopyFat16-4                2000000000          1.40 ns/op
BenchmarkCopyFat24-4                2000000000          1.40 ns/op
BenchmarkCopyFat32-4                2000000000          1.40 ns/op
BenchmarkCopyFat64-4                2000000000          1.40 ns/op
BenchmarkCopyFat128-4               2000000000          1.42 ns/op
BenchmarkCopyFat256-4               2000000000          1.41 ns/op
BenchmarkCopyFat512-4               2000000000          1.40 ns/op
BenchmarkCopyFat520-4               2000000000          1.40 ns/op
BenchmarkCopyFat1024-4              2000000000          1.40 ns/op
BenchmarkIssue18740-4                   100000         14531 ns/op
BenchmarkFinalizer-4                        500       2526732 ns/op
BenchmarkFinalizerRun-4                 1000000          2604 ns/op
BenchmarkSyscall-4                    20000000          63.4 ns/op
BenchmarkSyscallWork-4                20000000          96.9 ns/op
BenchmarkSyscallExcess-4              20000000          63.4 ns/op
BenchmarkSyscallExcessWork-4          20000000          96.8 ns/op
```

```
BenchmarkPingPongHog-4                        500000        2889 ns/op
BenchmarkStackGrowth-4                      100000000       21.2 ns/op
BenchmarkStackGrowthDeep-4                    500000        2548 ns/op
BenchmarkCreateGoroutines-4                  1000000        1046 ns/op
BenchmarkCreateGoroutinesParallel-4         10000000        198 ns/op
BenchmarkCreateGoroutinesCapture-4            200000        8941 ns/op          16 B/op
BenchmarkClosureCall-4                      300000000       5.80 ns/op
BenchmarkWakeupParallelSpinning/0s-4          100000       20843 ns/op
BenchmarkWakeupParallelSpinning/1µs-4          50000       23993 ns/op
BenchmarkWakeupParallelSpinning/2µs-4          50000       27880 ns/op
BenchmarkWakeupParallelSpinning/5µs-4          50000       37540 ns/op
BenchmarkWakeupParallelSpinning/10µs-4         30000       56729 ns/op
BenchmarkWakeupParallelSpinning/20µs-4         20000       96465 ns/op
BenchmarkWakeupParallelSpinning/50µs-4         10000       221160 ns/op
BenchmarkWakeupParallelSpinning/100µs-4         5000       359871 ns/op
BenchmarkWakeupParallelSyscall/0s-4            10000       321507 ns/op
BenchmarkWakeupParallelSyscall/1µs-4            5000       335555 ns/op
BenchmarkWakeupParallelSyscall/2µs-4            5000       346055 ns/op
BenchmarkWakeupParallelSyscall/5µs-4            5000       343007 ns/op
BenchmarkWakeupParallelSyscall/10µs-4           5000       386787 ns/op
BenchmarkWakeupParallelSyscall/20µs-4           5000       387292 ns/op
BenchmarkWakeupParallelSyscall/50µs-4           5000       556090 ns/op
BenchmarkWakeupParallelSyscall/100µs-4          3000       626904 ns/op
BenchmarkMatmult-4                          200000000       9.45 ns/op
BenchmarkFastrand-4                        2000000000       1.65 ns/op
BenchmarkFastrandHashiter-4                 50000000       32.3 ns/op
BenchmarkFastrandn/2-4                      200000000       8.42 ns/op
BenchmarkFastrandn/3-4                      200000000       8.43 ns/op
BenchmarkFastrandn/4-4                      200000000       8.50 ns/op
BenchmarkFastrandn/5-4                      200000000       8.42 ns/op
BenchmarkIfaceCmp100-4                        5000000       370 ns/op
BenchmarkIfaceCmpNil100-4                    10000000       203 ns/op
BenchmarkEfaceCmpDiff-4                       1000000       2052 ns/op
BenchmarkDefer-4                             5000000        256 ns/op
BenchmarkDefer10-4                          10000000        230 ns/op
BenchmarkDeferMany-4                         3000000        489 ns/op
BenchmarkRWMutexUncontended-4               20000000        112 ns/op
BenchmarkRWMutexWrite100-4                  10000000        197 ns/op
BenchmarkRWMutexWrite10-4                     1000000        1629 ns/op
BenchmarkRWMutexWorkWrite100-4              10000000        221 ns/op
BenchmarkRWMutexWorkWrite10-4                1000000        1314 ns/op
BenchmarkMakeSlice/Byte-4                   20000000       60.2 ns/op
BenchmarkMakeSlice/Int16-4                  20000000       73.2 ns/op
BenchmarkMakeSlice/Int-4                    10000000        133 ns/op
BenchmarkMakeSlice/Ptr-4                    10000000        188 ns/op
BenchmarkMakeSlice/Struct/24-4             10000000        200 ns/op
```

```
BenchmarkMakeSlice/Struct/32-4                  10000000         230 ns/op
BenchmarkMakeSlice/Struct/40-4                   5000000         262 ns/op
BenchmarkGrowSlice/Byte-4                        10000000         144 ns/op
BenchmarkGrowSlice/Int16-4                       10000000         204 ns/op
BenchmarkGrowSlice/Int-4                          5000000         245 ns/op
BenchmarkGrowSlice/Ptr-4                          5000000         365 ns/op
BenchmarkGrowSlice/Struct/24-4                    3000000         428 ns/op
BenchmarkGrowSlice/Struct/32-4                    3000000         486 ns/op
BenchmarkGrowSlice/Struct/40-4                    2000000         607 ns/op
BenchmarkExtendSlice/IntSlice-4                  10000000         165 ns/op
BenchmarkExtendSlice/PointerSlice-4              10000000         213 ns/op
BenchmarkExtendSlice/NoGrow-4                   100000000        15.4 ns/op
BenchmarkAppend-4                                30000000        56.2 ns/op
BenchmarkAppendGrowByte-4                             300     5560036 ns/op
BenchmarkAppendGrowString-4                             5   226765389 ns/op
BenchmarkAppendSlice/1Bytes-4                   100000000        12.6 ns/op
BenchmarkAppendSlice/4Bytes-4                   100000000        12.0 ns/op
BenchmarkAppendSlice/7Bytes-4                   100000000        11.9 ns/op
BenchmarkAppendSlice/8Bytes-4                   100000000        12.0 ns/op
BenchmarkAppendSlice/15Bytes-4                  100000000        11.9 ns/op
BenchmarkAppendSlice/16Bytes-4                  100000000        12.3 ns/op
BenchmarkAppendSlice/32Bytes-4                  100000000        12.6 ns/op
BenchmarkAppendSliceLarge/1024Bytes-4            1000000        1420 ns/op
BenchmarkAppendSliceLarge/4096Bytes-4             300000        4643 ns/op
BenchmarkAppendSliceLarge/16384Bytes-4            100000       13584 ns/op
BenchmarkAppendSliceLarge/65536Bytes-4             50000       32999 ns/op
BenchmarkAppendSliceLarge/262144Bytes-4            10000      120741 ns/op
BenchmarkAppendSliceLarge/1048576Bytes-4            2000      679068 ns/op
BenchmarkAppendStr/1Bytes-4                     100000000        13.5 ns/op
BenchmarkAppendStr/4Bytes-4                     100000000        11.9 ns/op
BenchmarkAppendStr/8Bytes-4                     100000000        11.9 ns/op
BenchmarkAppendStr/16Bytes-4                    100000000        12.6 ns/op
BenchmarkAppendStr/32Bytes-4                    100000000        14.4 ns/op
BenchmarkAppendSpecialCase-4                     20000000        85.0 ns/op
BenchmarkCopy/1Byte-4                           100000000        14.0 ns/op      71.25 MB/s
BenchmarkCopy/1String-4                         100000000        13.8 ns/op      72.29 MB/s
BenchmarkCopy/2Byte-4                           100000000        21.4 ns/op      93.67 MB/s
BenchmarkCopy/2String-4                         100000000        17.1 ns/op     117.24 MB/s
BenchmarkCopy/4Byte-4                           100000000        16.9 ns/op     237.34 MB/s
BenchmarkCopy/4String-4                         100000000        16.8 ns/op     237.50 MB/s
BenchmarkCopy/8Byte-4                           100000000        22.9 ns/op     348.61 MB/s
BenchmarkCopy/8String-4                         100000000        16.8 ns/op     475.00 MB/s
BenchmarkCopy/12Byte-4                          100000000        16.8 ns/op     712.43 MB/s
BenchmarkCopy/12String-4                        100000000        16.9 ns/op     712.04 MB/s
BenchmarkCopy/16Byte-4                           50000000        24.8 ns/op     646.37 MB/s
BenchmarkCopy/16String-4                        100000000        16.8 ns/op     949.90 MB/s
```

```
BenchmarkCopy/32Byte-4                                      100000000          16.3 ns/op  1964.35 MB/s
BenchmarkCopy/32String-4                                    100000000          16.4 ns/op  1956.17 MB/s
BenchmarkCopy/128Byte-4                                      50000000          27.4 ns/op  4676.48 MB/s
BenchmarkCopy/128String-4                                    50000000          27.4 ns/op  4676.64 MB/s
BenchmarkCopy/1024Byte-4                                     20000000           113 ns/op  9005.87 MB/s
BenchmarkCopy/1024String-4                                   20000000           113 ns/op  9004.83 MB/s
BenchmarkAppendInPlace/NoGrow/Byte-4                          1000000          1352 ns/op
BenchmarkAppendInPlace/NoGrow/1Ptr-4                           300000          4058 ns/op
BenchmarkAppendInPlace/NoGrow/2Ptr-4                           200000          7520 ns/op
BenchmarkAppendInPlace/NoGrow/3Ptr-4                           200000          8608 ns/op
BenchmarkAppendInPlace/NoGrow/4Ptr-4                           100000         14133 ns/op
BenchmarkAppendInPlace/Grow/Byte-4                            1000000          1076 ns/op
BenchmarkAppendInPlace/Grow/1Ptr-4                            1000000          1074 ns/op
BenchmarkAppendInPlace/Grow/2Ptr-4                            1000000          1449 ns/op
BenchmarkAppendInPlace/Grow/3Ptr-4                            1000000          1841 ns/op
BenchmarkAppendInPlace/Grow/4Ptr-4                            1000000          1983 ns/op
BenchmarkStackCopyPtr-4                                             5     309390029 ns/op
BenchmarkStackCopy-4                                                5     297313618 ns/op
BenchmarkStackCopyNoCache-4                                         2     518883532 ns/op
BenchmarkCompareStringEqual-4                                100000000          12.8 ns/op
BenchmarkCompareStringIdentical-4                           100000000          11.9 ns/op
BenchmarkCompareStringSameLength-4                          100000000          10.5 ns/op
BenchmarkCompareStringDifferentLength-4                    2000000000          1.40 ns/op
BenchmarkCompareStringBigUnaligned-4                             3000        430422 ns/op  2436.18 MB/s
BenchmarkCompareStringBig-4                                      3000        428766 ns/op  2445.59 MB/s
BenchmarkConcatStringAndBytes-4                              20000000          78.6 ns/op
BenchmarkSliceByteToString/1-4                              100000000          16.9 ns/op
BenchmarkSliceByteToString/2-4                               20000000          65.6 ns/op
BenchmarkSliceByteToString/4-4                               20000000          73.4 ns/op
BenchmarkSliceByteToString/8-4                               20000000          88.1 ns/op
BenchmarkSliceByteToString/16-4                              10000000           122 ns/op
BenchmarkSliceByteToString/32-4                              10000000           139 ns/op
BenchmarkSliceByteToString/64-4                              10000000           186 ns/op
BenchmarkSliceByteToString/128-4                             5000000           281 ns/op
BenchmarkRuneCount/lenruneslice/ASCII-4         50000000                        37.9 ns/op
BenchmarkRuneCount/lenruneslice/Japanese-4               5000000                 263 ns/op
BenchmarkRuneCount/lenruneslice/MixedLength-4           10000000                 232 ns/op
BenchmarkRuneCount/rangeloop/ASCII-4            50000000                        33.0 ns/op
BenchmarkRuneCount/rangeloop/Japanese-4                  5000000                 260 ns/op
BenchmarkRuneCount/rangeloop/MixedLength-4              10000000                 208 ns/op
BenchmarkRuneCount/utf8.RuneCountInString/ASCII-4  30000000                     41.4 ns/op
BenchmarkRuneCount/utf8.RuneCountInString/Japanese-4            10000000                 184 ns/op
BenchmarkRuneCount/utf8.RuneCountInString/MixedLength-4         10000000                 218 ns/op
BenchmarkRuneIterate/range/ASCII-4                              50000000                29.5 ns/
BenchmarkRuneIterate/range/Japanese-4                           5000000                 246 ns/op
BenchmarkRuneIterate/range/MixedLength-4                       10000000                 206 ns/op
```

35

```
BenchmarkRuneIterate/range1/ASCII-4                                    50000000              29.5 ns/
BenchmarkRuneIterate/range1/Japanese-4                                  5000000               246 ns/op
BenchmarkRuneIterate/range1/MixedLength-4                              10000000               221 ns/op
BenchmarkRuneIterate/range2/ASCII-4                                    50000000              29.5 ns/
BenchmarkRuneIterate/range2/Japanese-4                                  5000000               246 ns/op
BenchmarkRuneIterate/range2/MixedLength-4                              10000000               204 ns/op
BenchmarkArrayEqual-4                                                 200000000              6.33 ns
BenchmarkClockVDSOAndFallbackPaths/vDSO-4                              10000000               195 ns/op
BenchmarkClockVDSOAndFallbackPaths/Fallback-4                           1000000              1936 ns/op
BenchmarkTimeNow-4                                                     10000000               197 ns/op
PASS
ok      runtime     1331.420s
```

–Abishek Muthian (@heavyinfo)

Apple silicon

Cross-compiling Go applications

Host/Build: macOS/x86_64 Target: macOS/arm64 On host machine:

Xcode: Install latest Xcode which supports Apple silicon

Go: Install go*.darwin-amd64.pkg (or build go from source) on macOS/x86_64

Set go executable in PATH

Export the following environment variables before cross compiling go apps for macOS/arm64

```
export SDK_NAME=macosx
export SDK_PATH=$(xcrun --show-sdk-path --sdk $SDK_NAME)
export CGO_ENABLED=1
export GOOS=darwin
export GOARCH=arm64
export CC="$(xcrun -sdk $SDK_PATH --find clang) -arch $GOARCH -isysroot $SDK_PATH"
export CFLAGS="-isysroot $SDK_PATH -arch $GOARCH -I$SDK_PATH/usr/include"
export LD_LIBRARY_PATH="$SDK_PATH/usr/lib"
export CGO_CFLAGS="-isysroot $SDK_PATH -arch $GOARCH"
export CGO_LDFLAGS="-isysroot $SDK_PATH -arch $GOARCH"
Compile the app as below:
$ GOOS=darwin GOARCH=arm64 go build <app>
```

Copy the generated arm64 executable to target machine (macOS/arm64) and run