01 Go-- Overview
02 Language Features
03 Architecture
04 Demo
05 Evolution of Language
06 Future Work
01

Go-- Overview
Imperative, statically typed
Support for concurrency
Hide pointers
Hide memory management
C-like syntax
02

Language Features

Data Types and Data Structures

Gofunction

Channel
Data Types & Data Structures

Data Types:

- int: 4 bytes
- float: 8 bytes
- bool: 1 bit
- string: char *

Data Structures:

- Array: any 1 of data types + struct
- Struct: combination of any data types

```cpp
struct One{
    int x,
    float y,
    bool z,
    string r,
};

struct One x;
```

```cpp
function int main()
{
    array<int> a;
    a = new(array<int>[5]);
    a[0] = 4113;
    return 0;
}
```
Gofunction and threads

Lightweight concurrency

Thread pool implementation, initiated during first gofunction call

Same declaration as normal function with “gofunction” keyword

Calling function is easy, just like goroutine in golang.

Support up to three arguments and no return value

```
gofunction void add(int a, int b) {
    int i;
    int j;
    j = a + b;
    for (i=0; i < j; i = i + 1) {
        print(i);
    }
    return;
}

function int main() {
    int x;
    int y;
    int i;
    x = 5;
    y = 10;
    go add(x, y);
    go add(x+y, y);
    for(i = 0; i < 1; ){
    }
    return 0;
```
Channel

Can take in data of int, float, bool, string, and user-defined structs

Atomic and synchronized access to channel in a multi-threaded environment

Part of Control Flow

- Blocked on dequeue when reading from an empty channel
- Blocked on enqueue when writing into a channel at full capacity

Channel capacity can be determined at runtime

Suitable for producer-consumer paradigm
03

Architecture
04
Demo
/* mapper */
gap = range / threads;
for(i=0; i<threads; i = i+1)
{
    go search(words, i*gap, gap);
}

/* reducer */
go merge(threads * num_keys);

for(i = 0; i < num_keys; i++)
{
    reducer->tmp;
    prints("-----------");
    prints("word: " + tmp.k);
    print(tmp.count);
}
func search(arr []string, start int, range int) {
    i, j := 0, 0
    ret := make([]struct {kv} struct {kv}, num_keys)

    for i < num_keys; i++ {
        ret[i] = struct {kv} struct {kv} {keys[i], 0}
    }

    for i = 0; i < range; i++ {
        for j = 0; j < num_keys; j++ {
            if arr[start] == keys[j] {
                ret[j].kv.count++
                /* prints(keys[j]); */
                print(ret[j].kv.count) /*/ 
            }
        }
        start++
    }

    for i = 0; i < num_keys; i++ {
        ret[i]->mapper
    }
}
```c
function void merge(int num_reduce)
{
    int i;
    int j;
    array<struct kv> counts;
    struct kv tmp;
    counts = new(array<struct kv>[num_keys]);

    for(j=0; j<num_keys; j++){
        counts[j] = new(struct kv, keys[j], 0);
    }

    /* reduce */
    for(i=0; i < num_reduce; i++){
        mapper->tmp;
        for(j = 0; j < num_keys; j++){
            if(tmp.k == counts[j].k){
                counts[j].count = counts[j].count + tmp.count;
            }
        }
    }

    /* signal to main thread */
    for(j = 0; j < num_keys; j++){
        counts[j] -> reducer;
    }
}
```
Benchmark: Find-Waldo with Fib Calculation

m510: 8 Intel Xeon D-1548 at 2.0 GHz, 16 logic CPUs

Array of 1 million elements, find-waldo with 16 gocalls, dull-waldo with 1 main function

Total loads of calculation: fib(1) to fib(62500) calculations repeated 16 times + wordcount operations
Benchmark: No Threadpool vs Threadpool

m510: 8 Intel Xeon D-1548 at 2.0 GHz, 16 logic CPUs
100k operations
05

Evolution of Language
Iteration 1

Concurrent Threads
Channels with mmap
Function types (function, gofunction)
Structs
Arrays
First-Class Functions

Current Iteration

Concurrent threads
Function Types (function, gofunction)
Channel with malloc
Structs
Arrays
Future Work
Gocall
  Allowing gofunction with more than three arguments

Channel and Other Data Structures
  Adding a dictionary and nested dictionary/struct would be really helpful

Memory Usage
  Garbage collection

Input/Output
  File I/O, reading and writing from/to stdin/stdout/stderr