



D.E.M.E.T.R.A.

PSR Campania 2014-2020 - Sottomisura 16.1 Azione 2

Caserta – 31 Agosto 2020

APPENDICE A.03

Aspetti salienti di programmazione con Golang

PSR Campania 2014-2020 - Sottomisura 16.1 Azione 2 - Focus Area 3A

CUP: B28H19005170006



Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

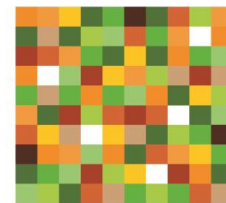
Unione Europea





Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

Unione Europea



PSR14-20
Campania

D.E.M.E.TRA.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAceability of the supply chain with blockchain technology

Il linguaggio di programmazione Golang è un progetto open source ed ha l'ambizione di rendere i programmatori più produttivi. Golang è stato realizzato dagli ingegneri di Google: Robert Griesemer, Rob Pike e Ken Thompson. La prima versione fu rilasciata come open source nel 2012.

Golang è un linguaggio compilato, progettato pensando alla programmazione dei sistemi. È fortemente tipizzato e sottoposto a garbage collection e ha un supporto esplicito per la programmazione simultanea. I programmi sono costruiti da pacchetti, le cui proprietà consentono una gestione efficiente delle dipendenze.

Golang è espressivo, conciso, pulito ed efficiente. I suoi meccanismi facilitano la scrittura di programmi che ottengono il massimo dalle macchine multicore e collegate in rete, mentre il sistema di tipo consente la costruzione di una programma flessibile e modulare. Golang si compila rapidamente in codice macchina, con una significativa potenza in fase di esecuzione.

INTRODUZIONE

La grammatica è compatta e semplice da analizzare, consentendo una facile analisi da parte di strumenti automatici come ambienti di sviluppo integrati.

GOLANG





Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

Unione Europea



PSR14-20
Campania

D.E.M.E.T.R.A.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAcability of the supply chain with blockchain technology

Sebbene Golang sia relativamente giovane rispetto agli altri linguaggi di programmazione, è stato già sviluppato sufficientemente e risulta così stabile da essere utilizzato dagli sviluppatori di tutto il mondo; persino lo stesso Google ricorre alle possibilità offerte da questo linguaggio di programmazione.

Golang si contraddistingue specialmente per la sua semplicità e multifunzionalità che gli sviluppatori avevano previsto sin dall'inizio. Infatti il nuovo linguaggio di programmazione nasce dall'insoddisfazione provocata dai rappresentanti affermati in questo settore che emergevano per una compilazione del codice efficiente o per una esecuzione veloce o ancora per un processo di programmazione semplice ma senza presentare tutti e tre i vantaggi insieme.

Tramite la combinazione di importanti feature tratte da diverse famiglie linguistiche, Golang dà in questo frangente il buon esempio: così la lingua di Google combina la facilità di programmazione, che contraddistingue il linguaggio interpretato e dinamico, con l'efficienza e la sicurezza di un linguaggio tipizzato e compilato in modo statico. In più la struttura di Golang mira a consentire un breve processo di sviluppo che permette di creare anche file grandi ed eseguibili su un singolo computer nel giro di pochi secondi.

CLASSIFICAZIONE

Alcune delle caratteristiche di markup decisive di Golang, tramite le quali può essere raggiunto questo standard di qualità, sono:

- un sistema di tipi con una forte proprietà di espressione ma leggero per una classificazione e differenziazione ottimale dei diversi oggetti (variabili, funzioni, ecc.);
- la concorrenza (programmazione parallela) per un'esecuzione più veloce del programma;
- la pulizia automatica (GC) per garantire un utilizzo ottimale della memoria disponibile e per evitare problemi correlati;
- una specificazione rigida di dipendenze senza una sintassi dichiarativa impegnativa;
- l'uso su tutte le piattaforme, consentendo così di utilizzare le applicazioni sviluppate su tutti i sistemi comuni.



Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

Unione Europea



D.E.M.E.T.R.A.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAceability of the supply chain with blockchain technology

La sintassi di Golang si basa sulla classica sintassi di C, ma si differenzia dal linguaggio di programmazione, già sviluppato nel 1972, con una serie di piccoli miglioramenti e una gamma di funzioni notevolmente ridotta. Così, ad esempio, nella programmazione con Golang non è obbligatorio inserire le parentesi tonde nelle condizioni e nei cicli e opzionalmente si può mettere il punto e virgola finale, tipico dei linguaggi della famiglia C.

In aggiunta si può regolare la validità degli identificatori (i nomi degli elementi suddetti) attraverso il modo di scrittura (maiuscolo o minuscolo). Ad esempio se un identificatore deve essere attivo anche al di fuori di un determinato pacchetto Golang, è necessario scrivere le prime lettere maiuscole. Di seguito vi elenchiamo altre particolarità della programmazione con Golang:

- Tutto il codice Golang comprensivo di dipendenze si può gestire tramite l'ambiente di lavoro GOPATH. La posizione di memorizzazione di questa directory obbligatoria GOPATH si può scegliere liberamente.
- I file sorgente su Golang si possono organizzare in modo modulare tramite directory che vengono indicate come packages o pacchetti. Il nome della relativa directory è così allo stesso tempo anche il nome del pacchetto, di cui fanno parte tutti i file sorgente che si trovano in questa cartella.
- Golang stabilisce determinate convenzioni per la formattazione del codice, ad esempio per l'intervallo esatto tra i singoli elementi. Quindi chi ha imparato a programmare applicazioni con Golang può anche leggere facilmente il codice di altri sviluppatori senza dover decifrare preventivamente il suo stile di formattazione personale, com'è il caso di molti altri linguaggi. Il formato non deve essere rispettato fin nel più piccolo dettaglio dal redattore: il tool integrato gofmt ottimizza automaticamente il codice Golang, risolvendo formattazioni errate.
- Tutti i file e i pacchetti che vengono importati nei progetti di Golang (autonomamente o da parte di terzi) sono sempre relativi alla cartella GOPATH/src rendendo così il processo di importazione molto facile. Inoltre Golang non compila gli elementi importati se non vengono effettivamente utilizzati. In questo modo un codice pulito è garantito solo quando i componenti importati non vengono o non sono più utilizzati.
- Con Golang si possono generare funzioni e metodi che possono restituire più valori. In questo modo ad esempio è possibile separare in maniera pulita un risultato valido e un errore inserito alternativamente al momento della restituzione.

PARTICOLARITÀ DI PROGRAMMAZIONE



Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

Unione Europea



D.E.M.E.TRA.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAceability of the supply chain with blockchain technology

TERMINOLOGIA DI BASE

PACCHETTI: i programmi Golang sono costituiti da "pacchetti", che rispecchiano i concetti di pacchettizzazione in altri linguaggi di programmazione (si pensi ai moduli in Python o ai pacchetti in Java). Ogni programma Golang contiene un pacchetto chiamato main, che funge da punto di ingresso del progetto.

MODULI: i moduli Golang sono librerie di terze parti installate da Golang. I moduli sono essenzialmente progetti che sono stati pubblicati per uso generale come dipendenze nei tuoi progetti.

FORNITORI: sebbene i moduli possano essere installati nella directory / pkg / mod per un uso globale, i progetti sorgente possono contenere le proprie versioni di questi moduli per evitare il conflitto di versioni di dipendenza tra i progetti (questo non è dissimile dagli ambienti virtuali Python). Sebbene non sia necessario, è possibile scegliere di mantenere le versioni del modulo specifiche del progetto.

Anche se superficialmente non sembra molto diverso dall'inizializzazione in C o C ++, l'inizializzazione in Golang è più potente. Durante l'inizializzazione è possibile costruire strutture complesse e le questioni di ordinamento tra oggetti inizializzati, anche tra pacchetti diversi, vengono gestite correttamente.

Le interfacce in Golang forniscono un modo per specificare il comportamento di un oggetto: se qualcosa può fare questo, allora può essere usato qui. Ad esempio: le stampanti personalizzate possono essere implementate con un Stringmetodo whileFprintf possono generare output per qualsiasi cosa con un Writemetodo. Le interfacce con solo uno o due metodi sono comuni nel codice Golang e in genere viene assegnato un nome derivato dal metodo, ad esempio io.Writer per qualcosa che implementa Write.

INIZIALIZZAZIONE e INTERFACCE

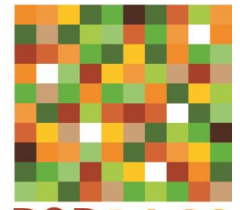
Un tipo può implementare più interfacce. Ad esempio, una raccolta può essere ordinata in base alle routine nel pacchetto sortse implementa sort.Interface, che contiene Len(), Less(i, j int) bool e Swap(i, j int), e potrebbe anche avere una formattazione personalizzata.

Se un tipo esiste solo per implementare un'interfaccia e non avrà mai metodi esportati oltre tale interfaccia, non è necessario esportare il tipo stesso. Esportare solo l'interfaccia rende chiaro che il valore non ha un comportamento interessante oltre a quanto descritto nell'interfaccia. Inoltre evita la necessità di ripetere la documentazione su ogni istanza di un metodo comune.



Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

Unione Europea



PSR 14-20
Campania

D.E.M.E.T.R.A.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAceability of the supply chain with blockchain technology

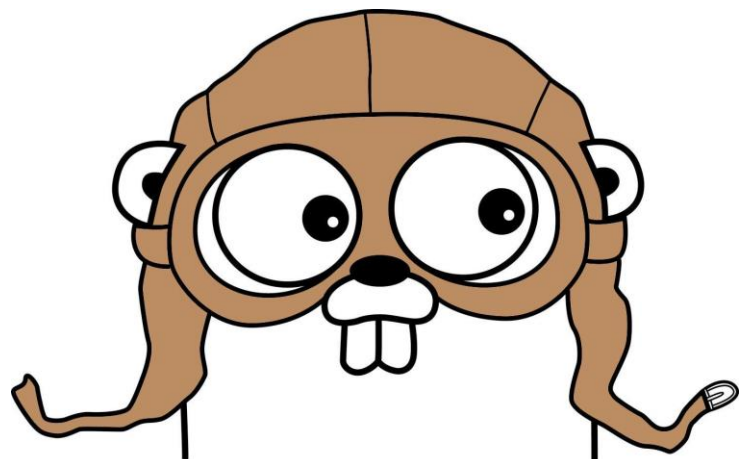
La programmazione simultanea in molti ambienti è resa difficile dalle sottigliezze richieste per implementare l'accesso corretto alle variabili condivise. Golang incoraggia un approccio diverso in cui i valori condivisi vengono trasmessi sui canali e, di fatto, mai condivisi attivamente da thread di esecuzione separati. Solo una goroutine ha accesso al valore in un dato momento. Non possono verificarsi gare di dati, in base alla progettazione. Per incoraggiare questo modo di pensare lo abbiamo ridotto a uno slogan: Non comunicare condividendo la memoria; invece, condividi la memoria comunicando.

Questo approccio può essere portato troppo lontano. Il conteggio dei riferimenti può essere fatto meglio inserendo un mutex attorno a una variabile intera, per esempio. Tuttavia, come approccio di alto livello, l'utilizzo dei canali per controllare l'accesso rende più facile scrivere programmi chiari e corretti.

Un modo per pensare a questo modello è considerare un tipico programma a thread singolo in esecuzione su una CPU. Non ha bisogno di primitive di sincronizzazione. Ora esegui un'altra istanza simile; anch'esso non necessita di sincronizzazione. Ora lasciate che quei due comunichino; se la comunicazione è il sincronizzatore, non c'è ancora bisogno di altra sincronizzazione. Le pipeline Unix, ad esempio, si adattano perfettamente a questo modello. Sebbene l'approccio di Golang alla concorrenza abbia origine nei processi CSP (Communicating Sequential Processes) di Hoare, può anche essere visto come una generalizzazione indipendente dai tipi di pipe Unix.

NON COMUNICARE
CONDIVIDENDO LA MEMORIA

CONDIVIDI LA MEMORIA
COMUNICANDO



D.E.M.E.TRA.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAceability of the supply chain with blockchain technology

Il Blockinterfaccia nel crypto/cipherpacchetto specifica il comportamento di un codice a blocchi, che fornisce la crittografia di un singolo blocco di dati. Quindi, per analogia con il bufiopacchetto, i pacchetti di cifratura che implementano questa interfaccia possono essere usati per costruire cifrari in streaming, rappresentati Streamdall'interfaccia, senza conoscere i dettagli della cifratura a blocchi.

ALGORITMI DI CIFRATURA

Ad esempio, nelle librerie hash sia `crc32.NewIEEE` che `adler32.New` restituiscono il tipo di interfaccia `hash.Hash32`. La sostituzione dell'algoritmo CRC-32 per Adler-32 in un programma Golang richiede solo la modifica della chiamata del costruttore; il resto del codice non è influenzato dal cambio di algoritmo.

Un approccio simile consente di cryptoseparare gli algoritmi di cifratura in streaming nei vari pacchetti dai codici a blocchi che concatenano.

VANTAGGI & SVANTAGGI

I punti di forza	I punti deboli
Multipiattaforma	Assenza di tipi generici
Sintassi molto semplificata	Orientata agli oggetti solo parzialmente
Gestione automatica del processo di memorizzazione (Garbage Collection)	Supporto di IDE (Integrated Development Environments) da potenziare
Formattazione unitaria del codice	Offerta proporzionalmente ridotta di librerie e pacchetti offerti da terze parti
Processo di importazione semplice	Passaggio faticoso dai linguaggi di programmazione classici, orientati agli oggetti, come Java e C++
Più valori di ritorno per funzioni e metodi	Per ora pochi tutorial, esperti, ecc.
Correzione del codice automatica (gofmt)	
Concorrenza	
Libreria standard ampia (per HTTP e compiti di rete)	



Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

Unione Europea



D.E.M.E.TRA.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAceability of the supply chain with blockchain technology

Questo report è stato realizzato con lo studio e l'approfondimento di testi scientifici, articoli e pubblicazioni redatti dai seguenti autori:

Riferimenti Bibliografici

<https://tour.golang.org>

https://golang.org/doc/effective_go.html

<https://golang.org/ref/spec>

Redattori del Report

<i>Contributi tecnici</i>	Fabio Fiori Thomas Chiozzi Michele Battagliola
<i>Editing testo</i>	Giuseppe Ciccarelli
<i>Supervisione</i>	Valeria Pucitti
<i>Approvazione</i>	Partner G.O. D.E.M.E.TRA.



Fondo europeo agricolo
per lo sviluppo rurale:
*l'Europa investe
nelle zone rurali*

Unione Europea



D.E.M.E.TRA.

Development of a tool to Evaluate the quantity of the cultivated product with satellite Monitoring of Earth for the TRAcability of the supply chain with blockchain technology



Report realizzato con il contributo del PSR Campania 2014-2020
Sottomisura 16.1 Azione 2 - Focus Area 3A