

# いいところ取りのほどよし コンパイル言語「Go」

土屋 健

## 特徴

### ● C/C++やJavaやPythonのいいところ取り

Go言語は、Googleが2009年に設計した新しいプログラミング言語です。2012年にバージョン1がリリースされました。Go言語は静的な型付きのコンパイル型言語で、オープンソースで開発されています。

Go言語が生まれたころは、CPUのマルチコア化が進み、クラウドといったネットワーク分散型の環境が一般的になりつつあったと思います。

マルチコア、ネットワーク環境で効率良く動作するアプリケーションを簡単に開発できる言語が必要になったのだと思われます。そして、Googleが自社のサービスの実装のために開発した言語ということもあり、簡潔に信頼性の高い並列処理が可能で、処理性能の高いプログラムが開発できます。

C/C++言語の処理性能、Javaのような(メモリ管理の観点での)信頼性や標準レベルでのパッケージ提供による開発効率の高さ、Pythonのような簡潔な記述力といった「それぞれのいいところ取りの言語」と言えます(後発の言語ですので、そういったメリットがないと意味がないですが…)。

以下にGo言語の特徴を挙げます。

### ● 処理性能が高い

ソースをコンパイルしてCPUネイティブ・コードに変換するため、CPUがフルスピードで動作します。

スクリプト言語やJavaなどの仮想マシン実行のプログラム言語と異なり、余分なランタイム環境を挟まず命令が直接CPUで実行されるため、オーバヘッドなしで実行されます。

### ● シンプル

言語仕様は小さく簡潔な記述、型の明確化によって記述の曖昧さを排除し、バグを作り込みにくくしています。また、言語自体の習得の敷居も低く、開発の現場では教育コストを抑えられます。

### ● 自動的なメモリ管理

ガベージ・コレクションを持ち、メモリ管理についてはコンパイラが自動的に最適な処理となるようにしてくれます。C/C++言語のようにメモリ獲得/解放を実装者が意識しなくてもよくなっています。

基本的にはポインタを使う必要はありませんが、ポインタを使った処理を書くこともでき、効率的な処理を記述できます。

ポインタを使用することでメモリ・コピーを防ぎ、余分なコピーの実行やメモリ消費を防げます(もっとも通常はGo言語に任せる方が効率が良いことが多いと思います)。

### ● 並列処理

goルーチンとチャンネルを使った並列処理の仕組みを持っています。goルーチンは関数を並列実行する仕組みで、関数をgo付き(go関数)で呼び出すことで、非同期で実行できます。非同期で実行すると、マルチCPUまたはマルチコア環境で複数の処理が同時に行えます。シングルCPUシングル・コア環境でも、IO処理待ちなどCPU処理以外の処理が行われている場合に、別の処理のCPU演算を同時に実行できます。

大抵のアプリケーション・プログラムでは、CPU演算よりもディスクやネットワークといったIO処理にかかる時間の方が大きいので、非同期処理による並列実行はプログラムの処理完了までの時間を短縮するために非常に有効です。

チャンネルは通信の仕組みでgoルーチン間での情報共有(受け渡し)や処理間の同期処理に利用します。

これらによって効率的な並列処理を簡単に実装できることが、サーバ開発で多く利用される理由でもあります。

### ● シングル・バイナリ

バイナリ・プログラムを1つ配置するだけで、ライブラリをあれこれ入れなくても動作させられます。C言語やJavaで開発された経験がある方はハマったことがあると思いますが、使用するライブラリやパッケージのバージョンが合わなかったり入手できなく