



時間短縮…ラズベリー・パイ2台で55%, 3台で40%に

ラズパイ・クラスタで実験… 分散コンパイル & ネットワーク・フート

ご購入はこちら

山田 英伸

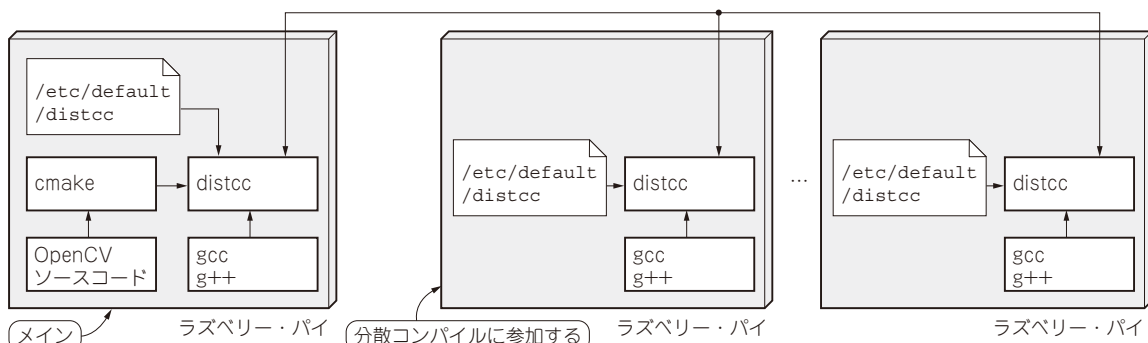


図1 複数のラズベリー・パイを使って分散コンパイルを行うシステムの構成

ラズパイを3台使ってアプリをビルドする

● ラズパイ上でのビルドは遅い

ラズベリー・パイを使って行う開発や検証作業においては、コンパイル不要なスクリプト言語 (Python など) だけでなく、コンパイルが必要なC/C++を使う場面も多くあります。幸いラズベリー・パイではセルフ・ビルドが可能なので、PCなどで使うLinuxの知識でC/C++がそのまま使えます。しかし、対象とするプログラム・サイズが大きくなってくると、コンパイルにかかる時間が長くなり、この作業がたらくってきます。

そこで、本稿では複数台のラズベリー・パイを用いた分散コンパイルによって作業時間を短縮してみます (図1)。

● 実際にビルド時間を測ってみる

今回は、本誌でもたびたび登場する画像処理ライブラリOpenCVのビルド完了までにかかる時間を測定しました。ビルド時間の測定に使った環境は次の通りです。

▶使用したソフトウェア

- Raspberry Pi OS Lite 64bit (2022-09-06)
- OpenCV 4.6.0
- gcc 10.2.1

▶使用したハードウェア

- ラズベリー・パイ4 8GBモデル
- ラズベリー・パイ4 4GBモデル (2台)
- QNAP製NAS (NFSサーバ、TFTPサーバとして使用)

ラズベリー・パイを使った分散コンパイルのシステムを写真1に示します。ネットワークでつながっているNAS (Network Attached Storage) の部分は含まれませんが、これ一式に、コンセントとLANケーブルをつなぐだけの取まりのよい形になっています。

▶ビルド時間の測定結果

使用するラズベリー・パイの数ごとのビルド時間を図2に示します。

ラズベリー・パイ1台で処理するよりも、複数台を使った方が処理に掛かる時間が短くなっています。2台を使って分散コンパイルをしたときには約半分の時間で処理が完了しており、3台を使うとさらに時間を短縮できました。

以降で、環境構築の方法と設定のポイントを解説します。

分散コンパイルの準備

- 分散コンパイルの処理はdistccにまかせる
分散コンパイルを実現するためにdistccという