

1 実験前の考察

ファイルシステムは OS とは独立しているが、現在多くの OS がサポートしている。これはメモリを分割してファイルとして割り当て、ディレクトリの構造の概念をメモリに持たせるものである。go のプログラムは package os を使用することで OS と連携し、ファイルへの書き込みを実現する。package os の File 構造体には Write 関数が設定されており、これは byte 型の引数に含まれる文字列を File 構造体で指定したファイルに書き込む。このとき引数とする byte 型で指定したデータを一気に OS に渡し、ファイル I/O 処理の回数を減らすのが golang のファイル書き込みにおけるバッファリングの手法である。このことから、バッファのサイズを大きくするごとに、ファイル I/O の回数はより減ることになり、処理時間はその分だけ短くなることと、ファイルサイズが大きくなるごとにその影響は顕著となることが予想できる。

今回の実験では、 $2^2 \sim 2^{12}$ のバッファサイズでデータファイルの大きさが 5MB になるように書き込みを行い、その処理時間を測る手順を 10 回繰り返した。その後、各バッファサイズについて処理時間の平均値を求め、1MB を書き込む秒数として処理速度を求めた。図 1 は、そのグラフである。

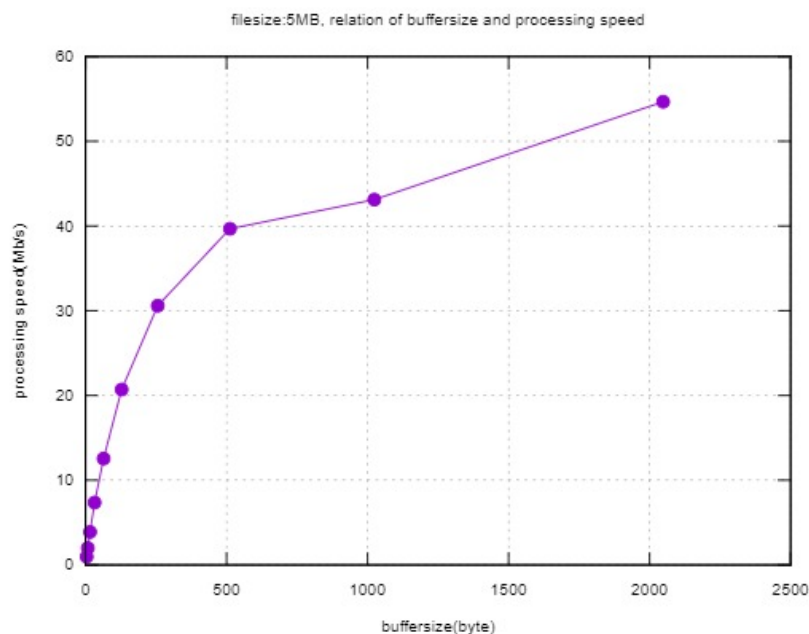


図 1 バッファサイズと処理速度の関連