

今回作成した Web アプリケーションプログラムに対して、各自で独自の改良を提案し実装せよ。

・ 改良点の説明

- 1.平均を求める。(回数に対しての平均 Mbits/sec を求めた)
- 2.グラフ化(実行回数分の数値をグラフ化した)

・ アピールポイント

- 1.平均(average.sh)

- A.grep などのコマンドを使い、数値だけ取得した。
- B.平均では「|bc」を使い、小数までの平均を求めた
- C.Mbits/sec や Kbits/sec に対応できるように単位を取得した。

- 2.グラフ化(gnuplot.sh) 図.2

- A.grep などのコマンドを使い数値を取得し、sed コマンドを使い回数をデータ(*.dat)に加えた。
- B.gnuplot を使いラベルなどを作成し、グラフ化した。

- 3.PHP(sample.php) 図.1

- A.「isset(\$a)」などを使い、正しく動くようにした。(入力されてるかどうかの判断)
- B.グラフのリンク先を貼った
- C.リンク先も入力されているかどうかの判断をさせた。

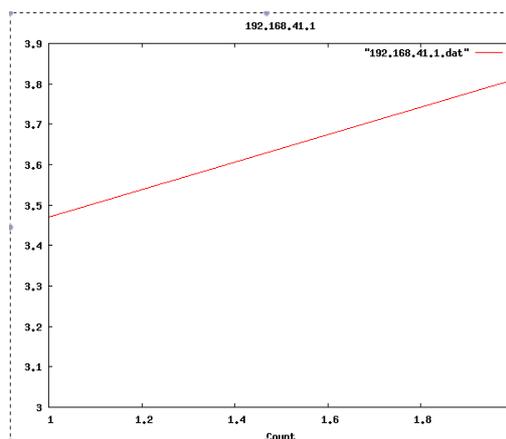
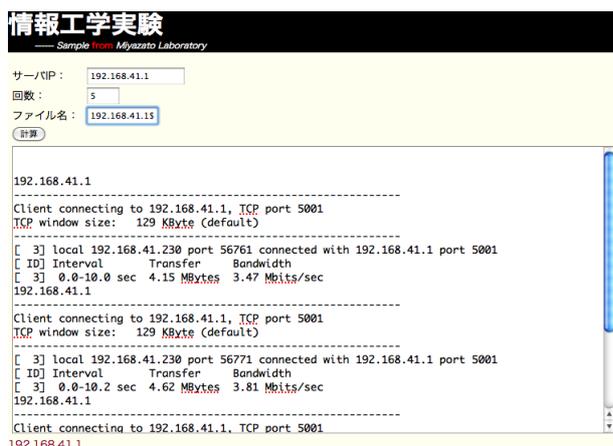


図.2 回数分のグラフ

・ 苦労したこと

- 1.iperfが絶対パスでなくてはならず、ずっと実行できなかった。
- 2.ファイルの出力がなかなかできなかった。Sampleのディレクトリの権限を変更するとできた。
- 3.phpの引数の与え方がわからなかった。シングルクォーテーション(')、またはドット(.)で分割するとできる
- 4.画像の出力でphpを使って出力をする。できたがファイルの数が増えたため、リンクだけ貼ることにした。
- 5.グラフ用のデータファイル(*.dat)を作るのが大変だった。データの数値だけでなく、回数をデータファイルに加えるために1行1行呼び出しをし、回数を加えていった。
- 6.変数をgnuplotに加える。\$fileを文字列と判断される。シングルクォーテーション(')でくることで、変数であると認識出来るようになった。
- 7.グラフを出力する時に文字列として表示された。グラフの表示をechoで出力させないようにした。リンク先をつくることで、グラフだけ差別化できるようになった。
- 8.Shellではpngを作ることができるのに、phpで出力すると作成させない。Gnuplotを絶対パスにすることで作成されるようになった。
- 9.phpで作成したファイルはroot権限でも削除することができなかった。php側にrmコマンドを記入した。