

DNAS アプリケーション
谷口 義樹

1 前回からの課題

- DNAS 上で動くランダムサーチの作成
- DNAS 上で動く ga の作成
- cvs のマニュアル改訂

Table 1 「TAG」と「DATA」

TAG	DATA
RSearch	01001010101111011010000011

2 進捗状況

2.1 DNAS 上で動くランダムサーチ

Distributed Network Application System(以下 DNAS) で用意されている API を用いて, DNAS システム上で動くアプリケーションであるランダムサーチを作成し, その DNAS アプリケーションの作成に必要なコンパイル方法や実行方法, 動作などをまとめた.

2.1.1 コンパイル

DNAS アプリケーションを作成し, コンパイルを行う場合は次のようにオプションを指定する必要がある (Fig. 1). これは, DNAS が提供するライブラリおよび, DNAS システムで用いられる pthread のライブラリを参照するために必要となる. 今回, DNAS を /home/yoshiki/local 以下にインストールしている. -ldnas-application, -lpthread はリンカオプション, -I,-L でそれぞれインクルードファイルとライブラリの場所を指定する.

```
$ gcc -ldnas-application -lpthread \
-I/home/yoshiki/local/include \
-L/home/yoshiki/local/lib \
-o random random.c
```

Fig. 1 コンパイル

2.1.2 実装

DNAS の提供する API を用いてランダムサーチを実装した. 対象問題は 30bit の OneMax 問題とした. 通信に使う関数は主に次の 2 つである.

DNAS_sendinfo(TAG,STRING,CDAT)

DNAS では「TAG」と「DATA」のペアを上流のノードに向けて送信する. 今回送信するのは, Table 1 に示すような「TAG」と「DATA」のペアになっている.

DNAS_gatherinfo(TAG,CDAT)

上流ノードの DNAS から, TAG が一致するデータを取得する.

2.1.3 実行結果

DNAS システム自体が起動されていない状態から, DNAS アプリケーションの実行するには, まず DNAS システム自身を起動し, それから指定のアプリケーションを実行する. さらに動作終了後, システム自身を終了させる. これら一連の流れを自作のシェルスクリプトに記述し, 動作させた. 8 ノードを用いて実行した結果, 次のような出力が得られた (Fig. 2). この結果から, 各ノードでランダムサーチが行われ, データが送受信されていることが分かる.

```
$ ./random-run.sh
dmachinemon-master v. 0.32.1 starting

(snip...)
amin-155: 000100100101000100111000111000
amin-155: current fitness : 11
amin-155: best fitness : 25

(snip...)
amin-150: 001110011111111101110100011011
amin-150: current fitness : 20
amin-150: best fitness : 24

Start cleaning:
(snip...)
```

Fig. 2 実行結果

3 翌月への課題

- DNAS 上で動く ga の作成
- dsh マニュアル改訂