

USP 研究所技術研究員
written by 大内智明

CODE Review

更新したファイルを相手サーバに配信する方法について説明します。

定期的に新しいデータを配信

業務で使用するデータは、日々更新されます。具体的
に言えば、小売り業界で使用する店舗や商品といったマ
スタ関連は、日に何度も新規登録・変更・削除といっ
た処理が繰り返されて、データ内容に更新が発生します。
ユニケーシでは、業務・データごとにサーバを分散配置^{※1}
することを推奨しているため、サーバ間でデータのやり取
りが行われます(図1)。IF するデータが大量にある場合
は、夜間処理だと全件の処理時間に問題がなくとも、日中
処理^{※2}では、IF するデータが更新される度に全件更新
を行うと処理時間に問題が出てくる場合があります。
こうした全件を処理して時間的な問題が発生する場合に
は、差分(未配信分)だけを配信することで処理時間を
短くできます。

今回は、即時反映^{※3}を実現するために、未配信分を
定期的に相手サーバに配信する常駐プログラムについて
の説明をします。

技術的な概要

常駐プログラム内で定期的に配信プログラムを実行

サーバ間で即時反映を行うためには、常駐プログラム
を実行して、定期的に未配信分データの有無を確認しま
す。未配信分があれば、配信を行います。

注意点として、常駐プログラムで無限ループが起きな
いように、終了条件のロジックを忘れずに盛り込む必要
があります(図2)。

【補足】

※1 業務ごとにサーバ群を配置して、役割を分散します。ただし、シス
テム負荷や業務用のデータ量が小さい場合には、1サーバ内に複
数の業務が混在することもあります。

※2 日中更新は、マスタ種類や業務内容によっては不要な場合もあ
ります。必要な場合は、業務内容により、即時反映を求められる場
合があります。

※3 サーバ間で配信を行うため、今回の例は完全な即時反映ではない。

図1 定期的に削除データを配信

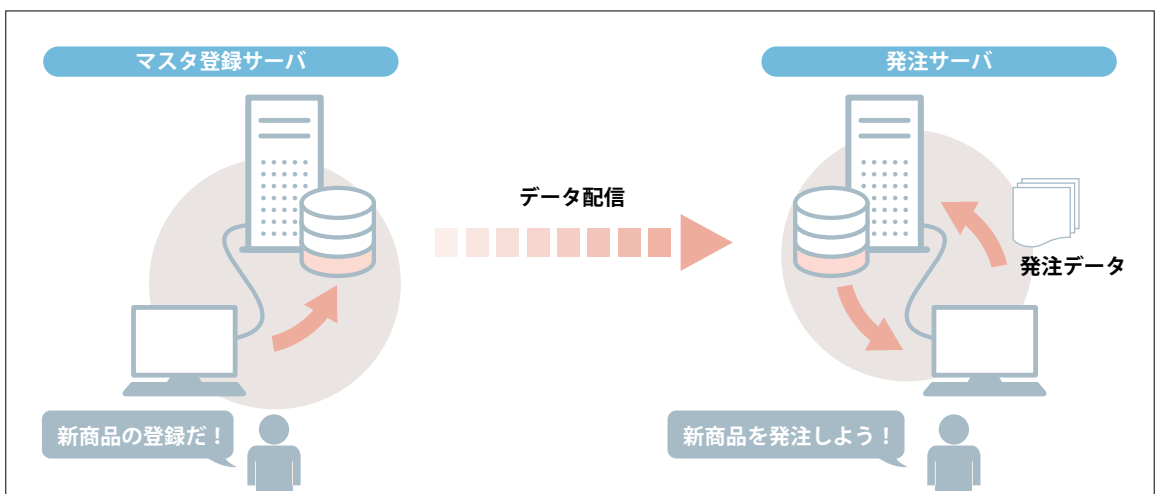
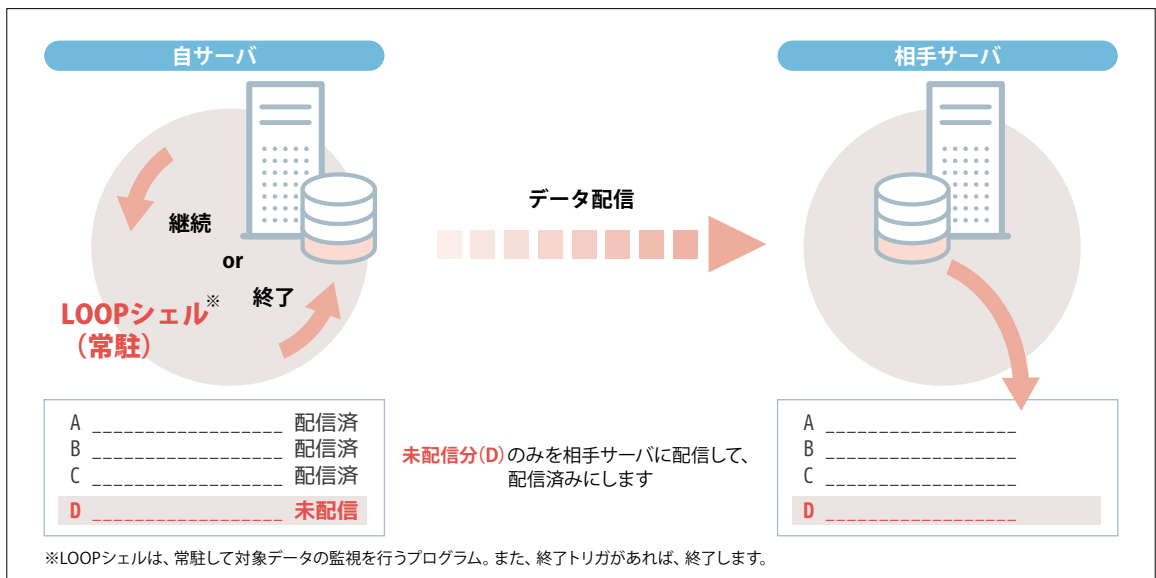


図2 未配信データの配信



リスト ファイル配信を行うかどうかを定期的に判断する処理

```

1 #!/usr/local/bin/bash -xv
2 # システム名: USPシステム
3 # サブシステム名: データ配信
4 # 業務名: LV3トランデータの配信
5 # プログラム名:
6 # 概要:
7 # 備考(Usage): DATAMASTER.LOOPREAL.LV3TRN_HAISIN <処理日付> <JOBグループ> <配信グループ> <インターバル>
8 # シェル名: DATAMASTER.LOOPREAL.LV3TRN_HAISIN
9 # 作成日: 20xx/xx/xx
10 # 会社名: USP
11 # 作成者: S.Yoshimura
12
13 #/////////////////////////////////////////////////////////////////
14 # 初期設定
15 #/////////////////////////////////////////////////////////////////
16
17 # 走行ログの記録
18 logd="${HOME}/LOG" # ログディレクトリ
19 logf="${logd}/LOG.${(basename $0)}.$(date +%Y%m%d)_%(date +%H%M%S)_" # ログファイル名
20 echo "${logf}" &> /dev/null
21 exec 2> ${logf} # 走行ログをファイルに出力
22
23 # パスの定義
24 PATH=/home/UTL:/home/TOOL:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:${PATH}
25 LANG=ja_JP.UTF-8
26
27
28 #-----
29 # 変数の定義
30 #-----
31 tmp="/tmp/${(basename $0)}_$(date +%Y%m%d%H%M%S)" # 一時ファイル
32 hostname="$(ghostname)" # サーバ名
33 semd="${HOME}/SEMAPHORE" # セマフォディレクトリ
34

```

```

35 # 引数設定項目のセット
36 sday=$1
37 myjob=$2
38 hgrp=$3
39 interval=$4
40
41 # その他変数定義
42 sysd="${HOME}/SYS"
43
44 # エラー時の終了処理定義
45 ERROR_EXIT(){
46     touch ${semd}/${(basename $0)_${myjob}_${hgrp}.${hostname}.ERROR.${sday}
47     echo "${hostname} ${(basename $0)_${myjob}_${hgrp}.${sday} ERROR $(date +%Y%m%d%H%M%S) ${logf}" >> ${logd}/UPCNT
48     exit 1
49 }
50
51 # 引数の確認
52 [ $# -ne 4 ] && ERROR_EXIT
53
54 # 簡易YYYYMMDD日付チェック
55 if ! isdate ${sday} ; then
56     echo "Parameter DATE error:[${sday}]"
57     ERROR_EXIT
58 fi
59
60
61 # 前回セマフォの消去
62 rm -f ${semd}/${(basename $0)_${myjob}_${hgrp}.${hostname}.*.${sday} >/dev/null 2>&1
63
64 # 起動時刻の記録
65 echo "${hostname} ${(basename $0)_${myjob}_${hgrp}.${sday} START $(date +%Y%m%d%H%M%S)" >> ${logd}/UPCNT
66 touch ${semd}/${(basename $0)_${myjob}_${hgrp}.${hostname}.START.${sday}
67
68
69
70 # 処理部
71 # 処理部
72 # 処理部
73
74 # 正マシンのみ処理
75 if [ "$(msctrl -host ${hostname} -job ${myjob} -print msflg)" = "M" ] ; then
76
77     # ループ処理開始
78     # ループ処理開始
79
80     while (sleep ${interval}) ; do
81
82         # 終了条件判定部
83         # 終了条件判定部
84
85         # 業務日付が変わったら終了
86         now_gyoumday=$(gyoumuday ${myjob} | self 1)
87         [ ${sday} != ${now_gyoumday} ] && break
88
89         # JOBGRPの夜間バッチ処理が開始になったら終了
90         [ $(gyoumuday ${myjob} | self 2) -ne 0 ] && break
91
92

```

処理日付

配信元JOBグループ

配信処理グループ

配信処理起動インターバル(秒指定)

シェルディレクトリ

isdate はユニケージコマンド
8桁日付のチェック。

本シェルの開始、正常終了、
異常終了ログを記録。

msctrl はユニケージコマンド
サーバの正副、役割、ホスト名などの
情報が取得できる。

self はユニケージコマンド
画面1参照。

役割のパラメータを引数に渡すと、戻り値に
1：本日の日付(yyyymmdd) 2：ステータス
が返ってきます。

```

93
94 #-----
95 # ループ内処理起動
96 #-----
97 # データ配信
98 shl=DISTRI_LV3TRN.DAY.HAISIN ※
99 ${sysd}/${shl} ${sday} ${myjob} ${hgrp} 1
100 [ $? -ne 0 ] && ERROR_EXIT
101
102 done
103
104 fi
105
106
107 #/////////////////////////////////////////////////////////////////
108 # 終了処理
109 #/////////////////////////////////////////////////////////////////
110 # 終了時刻の記録
111 echo "${hostname} ${basename $0}_${myjob}_${hgrp}_${sday} END $(date +%Y%m%d%H%M%S)" >> ${logd}/UPCNT
112 touch ${smd}/${basename $0}_${myjob}_${hgrp}.${hostname}.END.${sday}
113
114 # 終了
115 echo ${tmp}~* | xargs rm
116 echo "${basename $0} exit 0"
117 exit 0

```

※「DISTRI_LV3TRN.DAY.HAISIN」については、次号でご紹介します。

画面1 self

指定したフィールドのデータを取り出す

```

$ cat data
0000000 浜地_____ 50 F 91 59 20 76 54
0000001 鈴田_____ 50 F 46 39 8 5 21
0000003 杉山_____ 26 F 30 50 71 36 30
0000004 白土_____ 40 M 58 71 20 10 6
0000005 崎村_____ 50 F 82 79 16 21 80
$ self 4 2 data
F 浜地_____
F 鈴田_____
F 杉山_____
M 白土_____
F 崎村_____

```

まとめ

ユニケースでは、中規模以上の開発になると、業務・データ単位でサーバを分散配置することがあります。マスタ管理サーバでデータが更新されると、(業務によっては) 最新情報を更新する必要があります。マスタ種類と業務内容について理解を深めて、どのタイミングでマスタ配信が必要になるかを決めることは、システム負荷やネットワーク負荷を大きくしないためにも重要なポイントとなります。

コードの見どころ

- [1] 正サーバ上で実行する場合、本シェルが配信処理を行います。(①75行目)
- [2] 常駐プログラムの実行と継続・終了判定を行います。(②80～91行目まで、102行目)
- [3] データ配信用シェルを実行します。(③98～100行目まで)

