>Business made speedily

はやわかりLSF
【管理者編】

HPCシステムズ株式会社
2009/08/05 第2版

この資料の目的

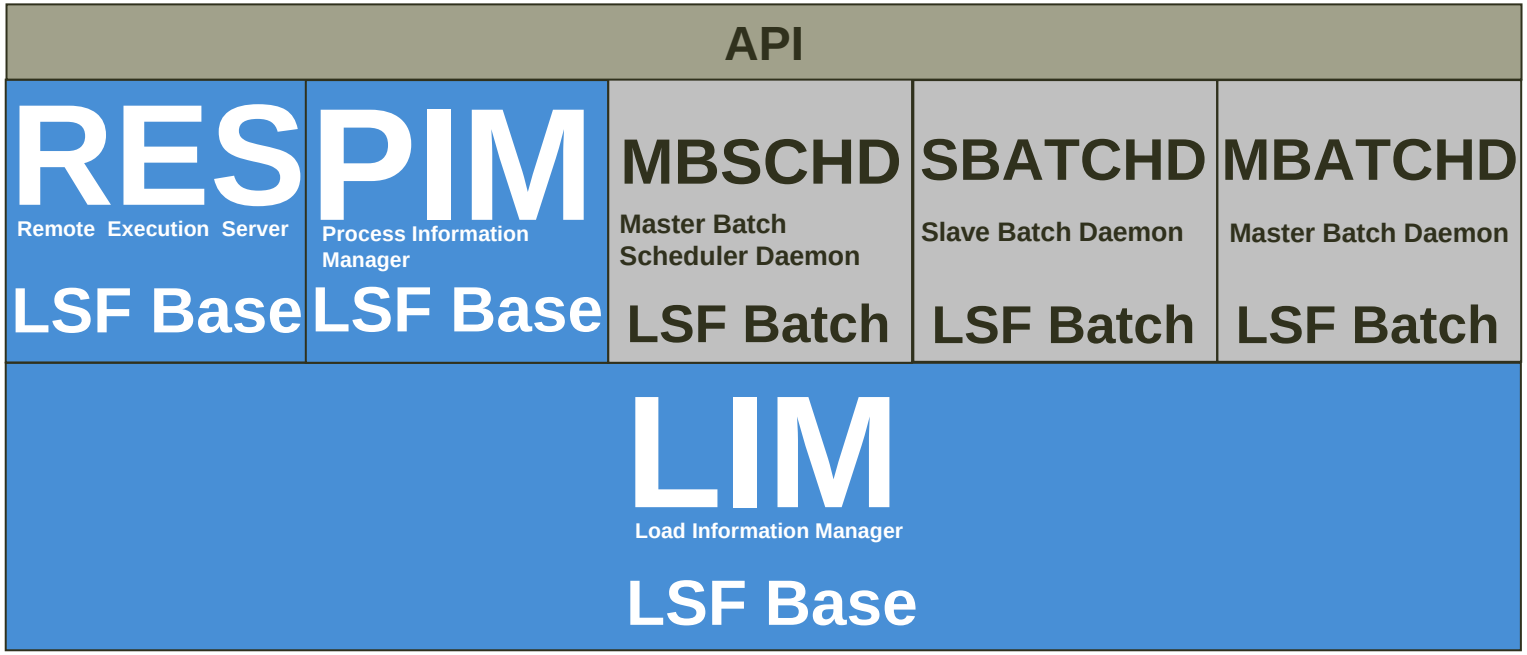
- LSF の動作原理（各種デーモン）を
  理解できるようになる
- LSF を安定して稼動し続けられるよう
  適切に管理できるようになる
- 効果的なキューの設定ができるようになる
- LSF の各種設定を変更・反映できるようになる

Agenda

- LSFのアーキテクチャ
  - LSFのソフトウェアアーキテクチャ
  - LSFで使われる用語の定義
  - 各種デーモン(lim, res, pim, sbatchd, mbatchd, mbschd)間の役割
  - LSFジョブライフサイクル
- LSFの設定
  - lim, mbatchd, mbschd の設定(lsf.conf)
  - lim構成ファイル(lsf.shared, lsf.cluster.<clustername>)
  - mbatchd構成ファイル(lsb.params, lsb.queues, lsb.hosts, lsb.resources, lsb.users, lsb.modules, lsb.serviceClasses)
  - キューの設定 (lsb.queues)
  - スケジューリングポリシ
    - FCFS(FIFO)
    - 優先割込み
    - フェアシェア
    - 排他
    - バックフィル
- LSFの管理
  - LSFデーモンの管理
  - キューのオープン/クローズ
  - ホストのオープン/クローズ
  - LSFのライセンス形態について

LSFのソフトウェアアーキテクチャ

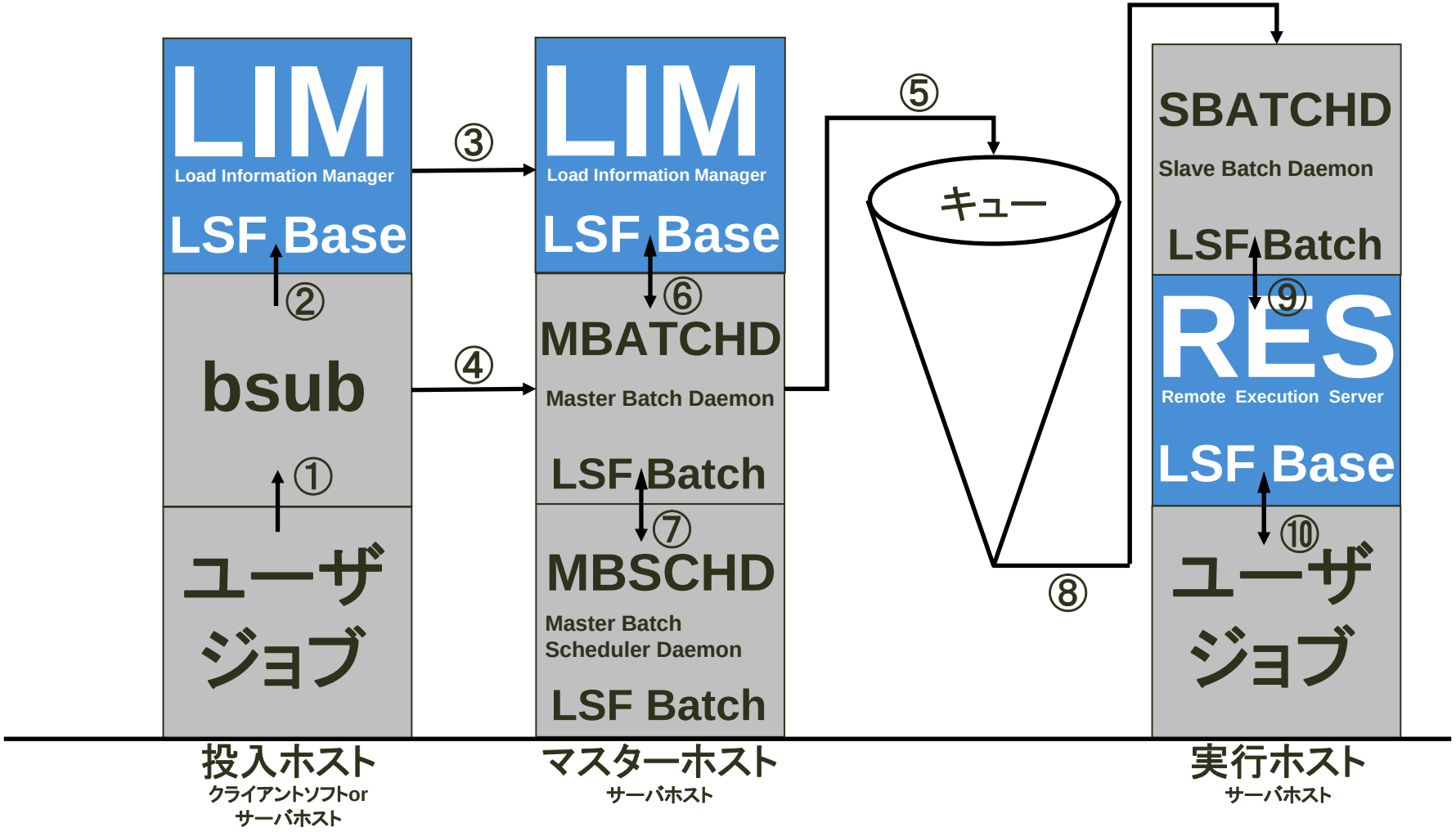
		bsub
		bjobs
		bhist
	lsadmin	bstop
	lshosts	bresume
	lsload	badmin
lsrun	lsid	bhosts



LSFで使われる用語の定義

- サーバホスト: ジョブを投入および実行する機能を持つホスト
- クライアントホスト: クラスタへのジョブの投入だけを行うことができるホスト
- マスタホスト: マスタ LIM と mbatchd を実行しているホスト

LSFの各種デーモン間の役割



LSFのアーキテクチャ::各種プロセス1

- sbatchd　　Slave Batch Daemon
  - クラスタ内の全てのサーバホスト上で実行される
  - mbatchdからJOB要求を受け取る②
  - 負荷閾値を執行する
  - サーバホストにおけるジョブの状態を保持する
  - マスターホストでmbatchdを起動する
- res　　Remote Execution Server
  - クラスタ内の全てのサーバホスト上で実行される
  - 対話型タスクを高速かつ透過的、安全にリモート実行する⑨
- lim　　Load Information Manager
  - クラスタ内のすべてのサーバホスト上で実行される
  - クラスタ構成を定義する
  - マスターホストを特定する
  - 組込み済みリソースの負荷情報を収集してマスターLIMに情報を送る
  - マスターELIM及びスレーブELIMにより収集されたサイト定義リソース負荷情報をマスターLIMにレポートする③
  - elim　　External Load Information Manager

LSFのアーキテクチャ::各種プロセス2

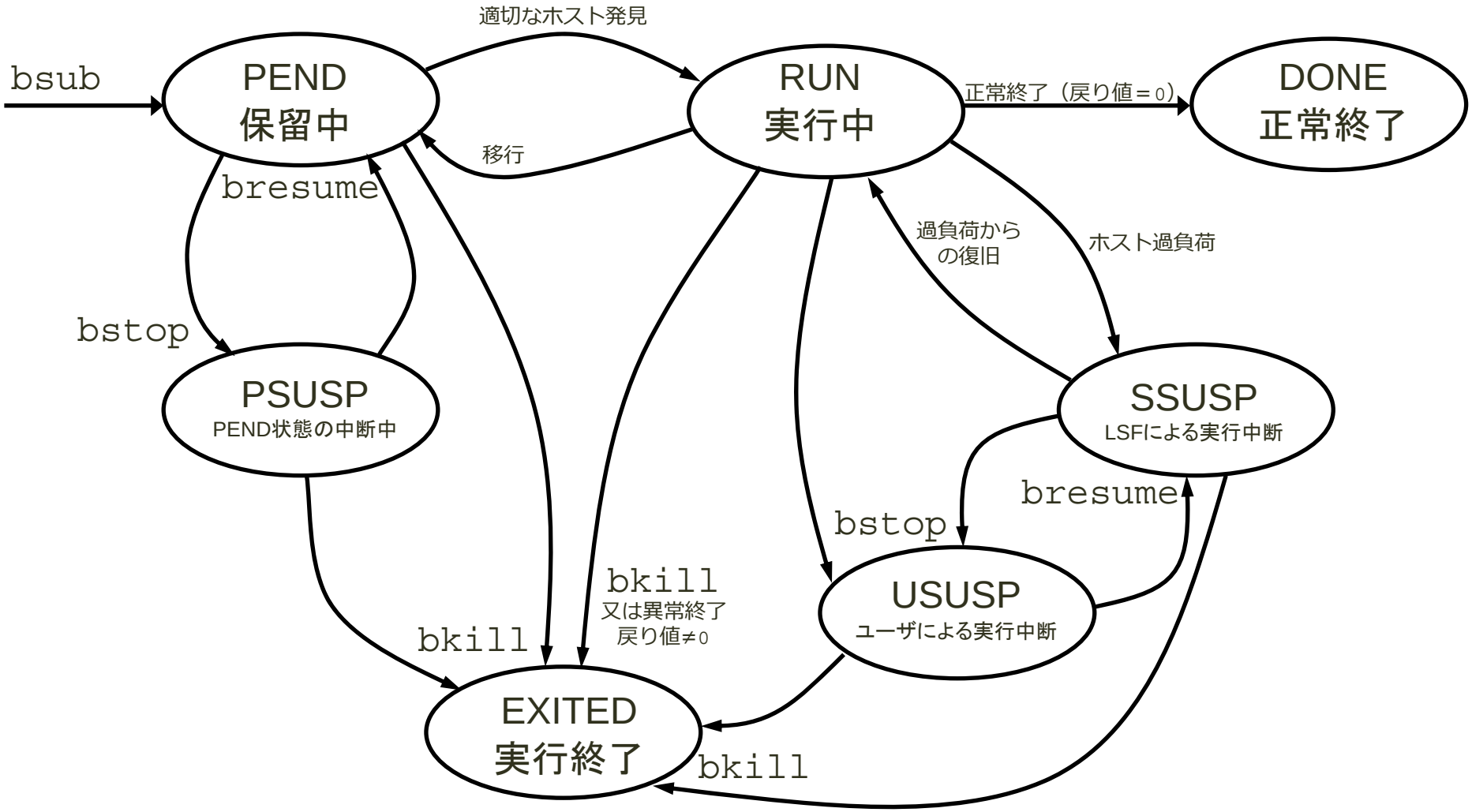
- pim　Process Information Manager
  - クラスタ内のすべてのサーバホスト上で実行される
  - サーバで実行されているジョブのプロセスを監視する
  - limによって自動起動される

  収集された情報は以下で使用される:
  - sbatchdで閾値を施行する時に使用される
  - mbatchdでフェアシェアを計算する時に使用される
- mbatchd　Master Batch Daemon
  - クラスタ毎に1つのmbatchdがマスターホスト上で実行される
  - ユーザからの問い合わせに応答する(bjobs,bhosts,等)
  - ジョブの要求を受け取る(bsub)④
  - バッチシステム内のすべてのジョブのライフサイクルでの状態に責任を持つ
  - マスターLIMから取得したリソースの負荷情報と保留しているジョブの情報をスケジューリングのためにmbschdに送る⑥
  - mbschdよりスケジューリングの決定を受けて⑦、指定されたサーバホストのsbatchdにジョブを実行する⑧
  - ジョブのトランザクションファイルを保持するキューを管理する⑤

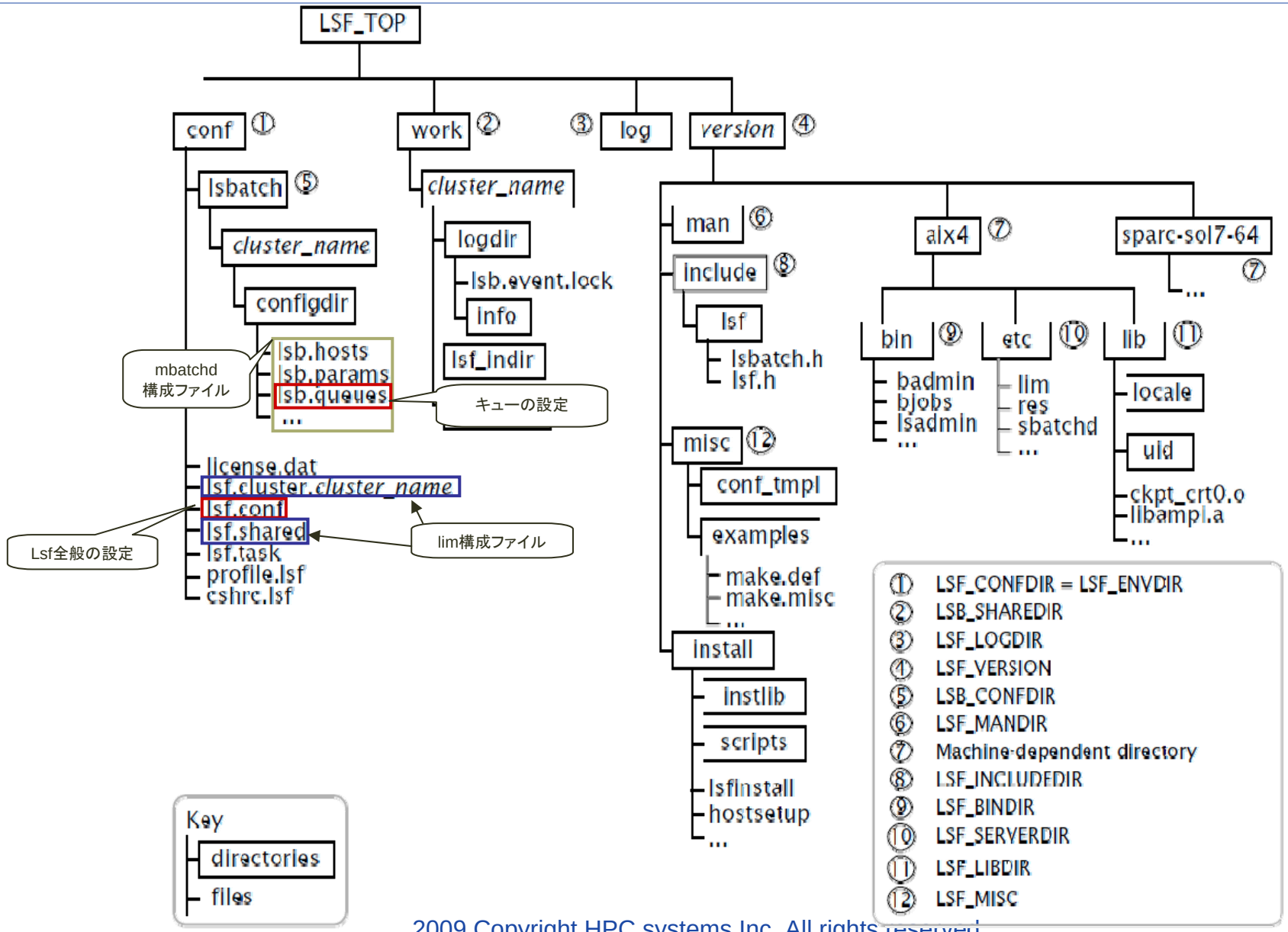
LSFのアーキテクチャ::各種プロセス3

- mbschd　　Master Batch Scheduler Daemon
  - クラスタ毎に1つのmbschdがマスターホスト上で実行される
  - mbatchdから保留ジョブの情報とリソース情報を受け取る
  - ジョブの要件やポリシーとリソースの空きを元に
    スケジューリングを決定する
  - ジョブの実行のために、スケジューリングの決定をmbatchdに送る⑦
  - lsf.confファイルから環境情報を読み取る

LSFジョブライフサイクル



LSFの設定::ディレクトリ構造



LSFの設定

- LSF全般の設定（ lim, mbatchd, mbschd の設定）★　　←今ココ
  - lsf.conf
- lim構成ファイル
  - lsf.shared
  - lsf.cluster.<clustername>
- mbatchd構成ファイル
  - lsb.params
  - lsb.queues
  - lsb.hosts
  - lsb.resources
  - lsb.users
  - lsb.modules
  - lsb.serviceClasses
- キューの設定
  - lsb.queues

LSF全般の設定::lsf.conf

- lsf.conf
  - lsf.conf ファイルはLSFのセットアップ時に作成され、LSF をインストールしたときに選択した、すべての設定がこのファイルに記録される。
  - lsf.conf 以外の各種 LSF 設定ファイルの格納場所およびLSFのサービスが設定される。
  - LSF デーモンや LSF の各種コマンドは、上述の設定を参照するためlsf.conf 内を検索
  - LSF を新しいバージョンにアップグレードした場合は、必要に応じて、lsf.conf が 更新される。

※=の前後に空白（スペース）を入れることは出来ません。

※値に複数の文字列を含めるには値を空白で区切り全体を”（ダブルクォーテーション）で括る必要が有ります。

LSF全般の設定::lsf.conf の例

```
# This file is produced automatically by lsfconfig
    according to
# installation setup. Refer to the "Administration
    Platform LSF"
# before changing any parameters in this file.
# Any changes to the path names of LSF files must be
    reflected
# in this file. Make these changes with caution.


LSB_SHAREDIR=/usr/share/lsf/work

# Configuration directories
LSF_CONFDIR=/usr/share/lsf/conf
LSB_CONFDIR=/usr/share/lsf/conf/lsbatch

# Daemon log messages
LSF_LOGDIR=/usr/share/lsf/log
LSF_LOG_MASK=LOG_WARNING

# Batch mail message handling
LSB_MAILTO=!U

# Miscellaneous
LSF_AUTH=eauth

# General lsfinstall variables
LSF_MANDIR=/usr/share/lsf/6.2/man
LSF_INCLUDEDIR=/usr/share/lsf/6.2/include
LSF_MISC=/usr/share/lsf/6.2/misc
XLSF_APPDIR=/usr/share/lsf/6.2/misc
LSF_ENVDIR=/usr/share/lsf/conf

# Internal variable to distinguish Default Install
LSF_DEFAULT_INSTALL=y

# Internal variable indicating operation mode
LSB_MODE=batch

# Other variables
LSF_LIM_PORT=6879
LSF_RES_PORT=6878
LSB_MBD_PORT=6881
LSB_SBD_PORT=6882

# WARNING: Please do not delete/modify next line!!
LSF_LINK_PATH=n

# LSF_MACHDEP and LSF_INDEP are reserved to maintain
# backward compatibility with legacy lsfsetup.
# They are not used in the new lsfinstall.
LSF_INDEP=/usr/share/lsf
LSF_MACHDEP=/usr/share/lsf/6.2

LSF_TOP=/usr/share/lsf
LSF_VERSION=6.2
LSB_SHORT_HOSTLIST=1
LSB_SUB_COMMANDNAME=Y
LSF_LICENSE_FILE=/usr/share/lsf/conf/license.dat
```

LSFの設定

- LSF全般の設定（ lim, mbatchd, mbschd の設定）
  - lsf.conf
- lim構成ファイル★　　←今ココ
  - lsf.shared
  - lsf.cluster.<clustername>
- mbatchd構成ファイル
  - lsb.params
  - lsb.queues
  - lsb.hosts
  - lsb.resources
  - lsb.users
  - lsb.modules
  - lsb.serviceClasses
- キューの設定
  - lsb.queues

lim構成ファイル::lsf.shared

- lsf.shared
  - クラスタ内のすべてのノードで共通の定義を設定
  - クラスタ名、ホストタイプ、ホストモデル、CPU係数、使用可能な特殊リソース、外部負荷インデックスのリストを定義
  - 現在 x86_64 OS を搭載した弊社出荷標準では、どのCPUを搭載しても、ホストタイプ:X86_64、ホストモデル: PC1133、CPU係数:23.1と認識される(`$ lshosts` の結果を確認すること)。これらは全て lsf.shared の一覧から選択されている。

※ PC1133 = PentiumIII Coppermine

lim構成ファイル::lsf.shared の例

```
# $Id: lsf.shared,v 5.84 2005/12/30 17:54:04 cchen Exp $

# ------------------------------------------------------------------------
# T H I S   F I L E:  Is shared by all clusters in the LSF system.
#
# This file contains all definitions referenced by individual
# lsf.cluster.<clustername> files. The definitions in this file can be
# a superset, i.e., not all definitions in this file need to be used in
# other files.
#
# See lsf.cluster(5) and "LSF User's and Admonistrator's Guide".
# ------------------------------------------------------------------------

Begin Cluster
ClusterName                     # Keyword
#white                          # Example
HPCS
End Cluster

Begin HostType
TYPENAME                        # Keyword
DEFAULT                         # used by lsfsetup
CRAYJ
ALPHA
HPPA
IBMAIX4
IBMAIX532
IBMAIX564
LINUX
LINUX2
LINUXAXP
LINUX86
LINUXPPC
LINUX64
HPUXIA64
MACOSX
LNXS39032
LNXS390X64
LINUXPPC64

(中略)
LINUX_ARM
X86_64
SX86_64
IA64
DIA64
SIA64
End HostType

#
# The CPU factor values are derived from SPECfp95
    given by hardware vendors
# or SpecBench (unless indicated otherwise)
# See http://www.specbench.org for more information on
    CPU benchmarking
# To find out an architecture string for a new model,
    run 'lim -t'
#
Begin HostModel
MODELNAME  CPUFACTOR   ARCHITECTURE # keyword
# x86 (Solaris, NT, Linux): approximate values, based
    on SpecBench results
# for Intel processors (Sparc/NT) and BogoMIPS results
    (Linux).
(中略)
PC1133                23.1
    (x6_1189_PentiumIIICoppermine)
(中略)
End HostModel

Begin Resource
RESOURCENAME  TYPE    INTERVAL INCREASING  DESCRIPTION
    # Keywords
(中略)
#  tmp2       Numeric 60       N           (Disk space
    in /usr/tmp in Mbytes)
#  nio        Numeric 60       Y           (Network
    I/O in Kbytes/second)
(中略)
End Resource
```

lim構成ファイル::lsf.cluster.<clustername>

- lsf.cluster.<clustername>
  - クラスタ定義情報 – 全てのLSFアプリケーションに影響を及ぼす。クラスタ管理者、キュー管理者やクラスタを構成するホスト、ホスト タイプやホスト モデルなどの個々のホストの属性、lsf.shared で定義したリソースを記述可能
  - LIMポリシー情報 – LIM のジョブ配置ポリシーに依存するアプリケーションに影響を与 えます。負荷の共有および LIM が提供するジョブ配置ポリシーを定義します。

lim構成ファイル::lsf.cluster.<clustername> の例

```
# $Id: TMPL.lsf.cluster,v 5.30 2005/03/15 16:36:30 lwang Exp $
#------------------------------------------------------------------------
# T H I S   I S   A   O N E   P E R   C L U S T E R   F I L E
#
# This is a sample cluster definition file.  There is a cluster
# definition file for each cluster.  This file's name should be
# lsf.cluster.<cluster-name>.
# See lsf.cluster(5) and the "Administering Platform LSF".
#

Begin   ClusterAdmins
Administrators = lsfadmin
End    ClusterAdmins

Begin   Host
HOSTNAME  model    type        server r1m  mem  swp  RESOURCES    #Keywords
#apple    Sparc5S  SUNSOL       1     3.5  1    2   (sparc bsd)   #Example
(中略)
#prune    !        !            1     3.5  1    2   (convex)
hpcs01  !       !       1       3.5 ()   ()   ()
hpcs02  !       !       1       3.5 ()   ()   ()
End     Host

Begin Parameters
LSF_HOST_ADDR_RANGE=*.*.*.*
PRODUCTS=LSF_Base LSF_Manager LSF_Sched_Fairshare LSF_Sched_Preemption LSF_Sched_Parallel LSF_Sched_Resource_Reservation
      LSF_Sched_Advance_Reservation Platform_HPC
End Parameters

# Begin ResourceMap
# RESOURCENAME  LOCATION
# tmp2          [default]
# nio           [all]
# console       [default]
# End ResourceMap
```

LSFの設定

- LSF全般の設定（ lim, mbatchd, mbschd の設定）
  - lsf.conf
- lim構成ファイル
  - lsf.shared
  - lsf.cluster.<clustername>
- mbatchd構成ファイル★　　←今ココ
  - lsb.params
  - lsb.queues
  - lsb.hosts
  - lsb.resources
  - lsb.users
  - lsb.modules
  - lsb.serviceClasses
- キューの設定
  - lsb.queues

mbatchd構成ファイル

- mbatchd構成ファイル
  - lsb.params
    - デフォルトキュー
    - sbatchdスケジューリング間隔
    - 履歴時間(CLEAN_PERIOD)
    - 割込間隔 etc.
  - lsb.queues
    - キュー　※詳細は後ほど
    - スケジューリングポリシー
    - 優先順位
    - ジョブの制御方法
  - lsb.hosts
    - 最大ジョブスロット(MXJ)
    - ディスパッチウィンドウ
    - ホストパーティション
    - ホストグループ
  - lsb.resources
    - リソース制限
    - コンシューマ制限
  - lsb.users
    - ユーザグループ
    - 階層的フェアシェア
  - lsb.modules
    - スケジューリングのプラグイン
  - lsb.serviceClasses
    - サービスレベル保証定義(SLA)

LSFの設定

- LSF全般の設定（ lim, mbatchd, mbschd の設定）
  - lsf.conf
- lim構成ファイル
  - lsf.shared
  - lsf.cluster.<clustername>
- mbatchd構成ファイル
  - lsb.params
  - lsb.queues
  - lsb.hosts
  - lsb.resources
  - lsb.users
  - lsb.modules
  - lsb.serviceClasses
- キューの設定★　　←今ココ
  - lsb.queues

キューの設定::lsb.queues 1

- lsb.queues
  - 各キュー定義は Begin Queue 行から始まり、End Queue 行で終わります。
  - ADMINISTRATORS
    - キュー管理者は、キュー自体はもちろん、キュー内の全てのユーザのジョブに対して操作を行うことができます。
  - BACKFILL
    - = Y or N キューに対してバックフィル スケジューリングが有効 になります。
  - CORELIMIT
    - このキューに投入されたジョブに属する全てのプロセスに対する、プロセスあたりのコアファイルサイズ制限（単位:KB）。
  - CPULIMIT
    - ジョブが使用できる CPU 時間の合計を制限します（単位:分）。ジョブ全体の CPU 時間の合計が制限に達すると、ジョブの実行が中止されます。
  - DISPATCH_WINDOW
    - このキューのジョブを実行する時間帯域。実行終了時刻を過ぎても、一度投入されたジョブは動き続けます。
  - DESCRIPTION
    - `bqueues -l` によって表示されるキューの説明。
  - EXCLUSIVE
    - = Y or N 排他キューを指定します。本設定のされているキューに対して`bsub -x` で投入された排他 JOB のみ排他の対象となります。
  - FAIRSHARE
    - キューレベルのフェアシェアを有効にし、シェアの割り当てを指定します。スケジューリングポリシについては、後述します。
  - FAIRSHARE_QUEUES
    - キュー間フェアシェアを定義します。
  - HOSTS
    - このキューに投入されたジョブが実行できるホストのリストです。
  - INTERACTIVE
    - = NO or ONLY このキューに対話型ジョブを却下させるか(NO)、対話型ジョブだけを受け付ける(ONLY)設定ができます。

キューの設定::lsb.queues 2

- lsb.queues
  - MEMLIMIT
    - このキューに投入されたジョブに属するすべてのプロセスに対する、プロセスあたりの（ハード）サイズ制限（単位: KB）。有効にするには、lsf.conf の LSB_MEMLIMIT_ENFORCE=y 又は LSB_JOB_MEMLIMIT=y に設定しておく必要が有ります（詳しくは lsf.conf の man を読むこと）。
  - NEW_JOB_SCHED_DELAY
    - 新しいジョブが、スケジュールされるまで待機する秒数。値ゼロ(0)はジョブが直ちにスケジュールされることを意味します。
  - PREEMPTION
    - = PREEMPTIVE / PREEMPTABLE 割り込みスケジュールを有効にし、キューに割り込みポリシを定義します。
  - PRIORITY
    - キューの優先順位。値が大きいほど、他のキューと比較した場合のジョブ実行優先順位が高くなります。
  - PROCLIMIT
    - ジョブに割り当て可能なスロットの最大数。並列ジョブでは、ジョブに割り当て可能な プロセッサの最大数。
  - RES_REQ
    - 有効なホストを決定する為に使用されるリソース要件設定。
  - RESUME_COND
    - 負荷が指定した閾値を満たした場合、LSFはこのキューで中断されていた(SSUSP)ジョブを再開します。
  - RUN_WINDOW
    - キュー内のジョブが実行可能な期間の設定。ウィンドウが閉じられると、LSFはキューで実行中のジョブを中断します。
  - STOP_COND
    - 負荷が指定した閾値を満たした場合、LSFはこのキューで実行中のジョブを中断します。
  - RUNLIMIT
    - 実行制限時間の最大値とオプションとしてデフォルト実行制限値を指定します。デフォルトでは、指定した実行制限の最大値よりも長時間 RUN 状態になっているジョ ブは、LSF によって強制終了されます（単位:分）。
  - USERS
    - このキューにジョブを投入可能なユーザのリストです。

mbatchd構成ファイル:: lsb.queuesの例

```
Begin Queue
QUEUE_NAME     = priority
PRIORITY       = 43
NICE           = 10
FAIRSHARE     = USER_SHARES [[default,1]]
SLOT_RESERVE = MAX_RESERVE_TIME[15]
PREEMPTION = PREEMPTIVE
DESCRIPTION    =  Jobs submitted for this queue are
  scheduled as urgent¥
jobs. Jobs in this queue can preempt jobs in lower
  priority queues.
End Queue
```

キューの設定::キューの無効化

- ファイルからキューとその設定を削除する
- Begin および End Queue 行をコメントアウトする
- キュー構成をコメントアウトする必要はない

```
# Begin Queue
QUEUE_NAME = interactive
DESCRIPTION = default for interactive jobs
ADMINISTRATORS = user2 userGroupA
PRIORITY = 80
INTERACTIVE = ONLY
NEW_JOB_SCHED_DELAY = 0
HOSTS = hostGroupB+5 hostGroupA+2 others
# End Queue
```

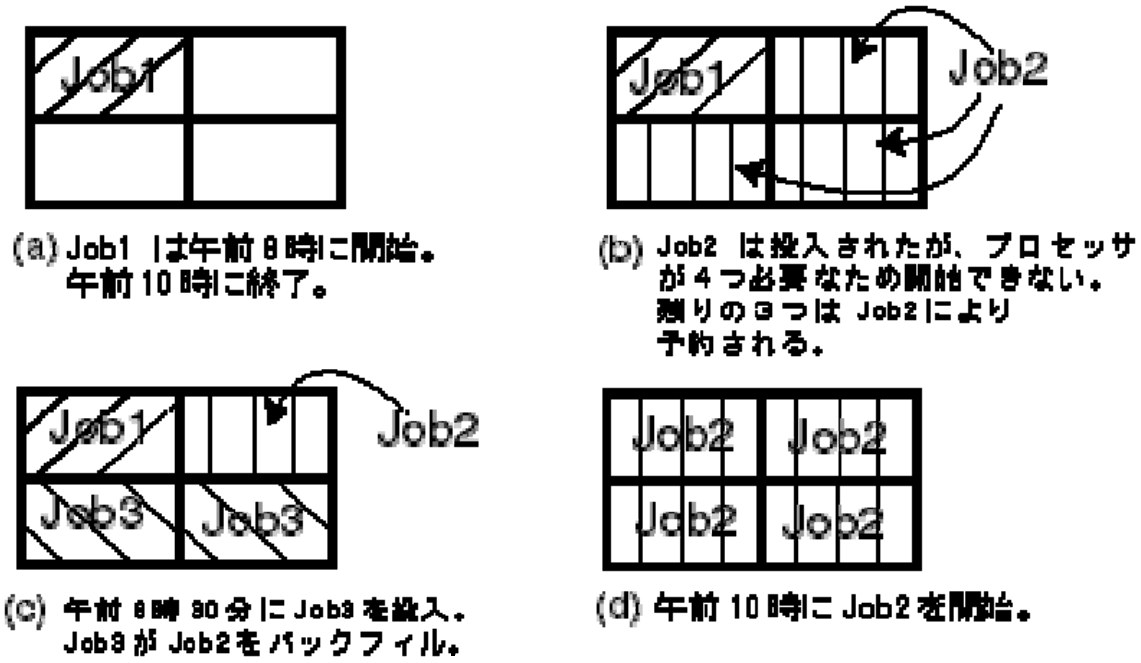
LSFの設定::スケジューリングポリシ 1

- スケジューリングポリシ
  - FCFS
    - ジョブをキューの中での順番どおりに実行しようとする
    - LSFの標準設定（ただし、弊社では工場出荷時**フェアシェア**に設定）
    - first-come, first-served（日本語では「先着順」、「早い者勝ち」）
  - 優先割込み
    - 保留中の優先順位の高いジョブが、実行中の優先順位の低いジョブから リソースを取り上げる
  - フェアシェア
    - リソース(CPU時間,メモリ)の使用量が少ないほど優先順位が高くなります。
    - 複数人数での使用に適した設定ですが、ジョブが投入した順番に実行されないため、 ユーザが意図しない順番でジョブが実行されることがあります。
  - 排他
    - 特定のジョブに実行先のホストを排他的に使用させることができます。

LSFの設定::スケジューリングポリシ 2

- バックフィル
  - バックフィル スケジュール機能を使用すると、大型ジョブが開始する前に実行を完了できるような小型のジョブが、予約済みのジョブ スロットを使用できるようになります。この機能により、リソースの使用率が向上するため、LSF のパフォーマンスが改善されます。

  - バックフィルを使わない場合、大型並列ジョブを早急にスケジュールするためにプロセッサ予約を利用したとします。予約済みのプロセッサは、そのジョブが開始されるまでアイドル状態になっています。短い時間内に終わるジョブが投入されても、キューに溜まる一方です。



LSFの管理

- LSFの管理
  - LSFデーモンの管理
    - lsb.* ファイルの変更反映

      ```
      # badmin mbdrestart
      ```
    - lsf.* ファイルの変更反映

      ```
      # lsadmin reconfig
      # badmin mbdrestart
      ```
  - キューのオープン/クローズ

    ```
    # badmin qopen queuename  ジョブの投入を受け付ける
    # badmin qclose queuename  ジョブの投入を受け付けない
    # badmin qact queuename  キューからジョブを実行する
    # badmin qinact queuename  キュー内のジョブを実行しない
    ```
  - ホストのオープン/クローズ

    ```
    # badmin hopen hostname   ホストはジョブを受け付ける
    # badmin hclose hostname  ホストはジョブを受け付けない
    ```

LSFのライセンス形態について

- 弊社で主に取り扱うライセンスは次の3種類
  - デモライセンス(MACアドレス不問、コア数不問、ただし短期間で失効)

```
FEATURE lsf_base lsf_ld 6.200 23-JUN-2009 0 0A1BC234DF56GH7I8901 "Platform" DEMO
```

  - レンタルライセンス
    (eth0のMACアドレス固定、コア数固定、有効期限以降失効)

```
SERVER hpcs01 00102AB304CD 1700
INCREMENT lsf_base lsf_ld 6.200 29-APR-2014 2 0A1BC234DF56GH7I8901 VENDOR_STRING="HPC Systems" ISSUED=29-MAR-2009
NOTICE="PARTNER S-Class" SN=E12345
```

  - 永久買い取りライセンス
    (eth0のMACアドレス固定、コア数固定、失効しない)

```
SERVER hpcs01 00102AB304CD 1700
INCREMENT lsf_base lsf_ld 6.200 1-jan-0000 2 0A1BC234DF56GH7I8901 VENDOR_STRING="HPC Systems" ISSUED=29-MAR-2009
NOTICE="PARTNER S-Class" SN=E12345
```

- ライセンスファイルは、
  `/usr/share/lsf/conf/license.dat`
  に格納されている

※eth0 とは LAN ポートのうち、Linux が最初に認識するポート
※MAC アドレスとは LAN ポートに固有で割り振られる(通例)一意の英数字