>Business made speedily

はやわかりLSF
【利用者編】

HPCシステムズ株式会社
2009/10/01 第5版

この資料の目的

- LSF とは何か
- LSF を使うメリット
- ジョブ管理システムに LSF を選ぶ利点

  について、的確に説明できるようになる

- LSFでJOBを投入できるようになる
- 投入したJOBの実行状況を把握できるようになる

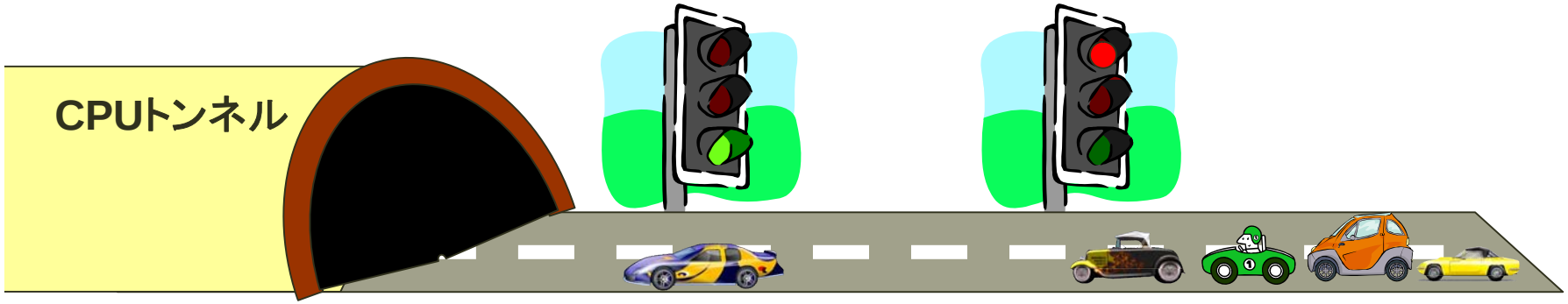
Agenda

- LSFについて
- ジョブ管理システムにLSFを選ぶ利点
- 他のジョブ管理システムと LSF の違い
- LSFで動作実績のあるアプリケーション
- LSFの使い方

LSFは計算の交通整理をするソフトウェアです

- CPUに計算を大量に投入すると、渋滞が発生し、通過（実行終了）までに通常より時間が掛かってしまうことがあります。

- LSF はCPUに入る計算量を調整する、交通信号の役割を果たします。



LSFについて::バッチジョブとは？

- 計算の実行方式
  - インタラクティブジョブ
    - Linux のコマンドのように、コマンドを入力したら、すぐに結果が返ってくるようなジョブのこと。
    - **開始時間は今、ログインしているノードで**実行される。
    - ノードがCPUやメモリを100%使い切っている状態でも、無理に流れようとする。→ジョブが異常終了する原因になる。
  - バッチジョブ
    - 処理の手順やデータを与えておいて一括して処理させる方式。
    - **いつ、どのノードで**動作させるかは、**ジョブ管理システム**(LSF)が、空いているノードや時間を自動的に(適切に)計画し、実行してくれる。

LSFについて::ジョブ管理システムとは

1. **リソース**（計算機またはCPU,メモリ,HDD等）
   管理機能

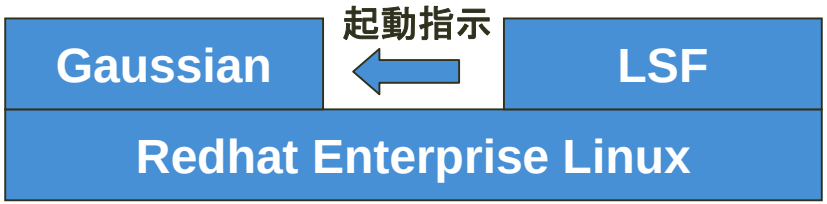
2. **ジョブ**(ユーザがジョブ管理システムに投入した
   コマンド)を**受け付ける機能**

3. ジョブを、**いつどこで開始するか計画**し、
   実行し、**結果を記録・報告**する機能

以上　3つの機能を備えたソフトウェアのこと。

LSFについて::そもそも、LSFとは？

- Load Sharing Facility（負荷分散機構）の略
- ジョブ管理システムというジャンルに属する
- OS=基本ソフトウェア（例えば Redhat Enterprise Linux）の上で動き、ユーザのジョブ(例えば Gaussian) を計画し、実行するアプリケーションソフト



- Platform Computing Inc. 製
  - 本社:カナダ
  - 日本:プラットフォームコンピューティング株式会社

LSFは業界標準のジョブ管理システムです

- Platform 社製品を採用している海外の企業様
  - 電子: AMD, Broadcom, シスコ(Cisco), テキサス・インスツルメンツ (TI).....
  - 金融: プルデンシャル, JPモルガン・チェース, KBCファイナンシャル.....
  - 製造業: エアバス, ボーイング, ゼネラル・エレクトリック(GE).....
  - エネルギー: シェル, アジップ, BP, クウェート国営石油開発, トタル.....
  - 政府・教育: 欧州原子核研究機構(CERN), アメリカ国防総省(DoD).....
  - 製薬: Johnson & Johnson, ノバルティス(Novartis), ファイザー(Pfizer).....

- LSF のシェアは国内でも650社以上
  主要企業様や政府系研究機関様をほぼ網羅！

- 大手計算機センターと同じ環境がお手元に！

LSF活用事例::人が2人集まれば・・・・・・

- 同じフロアで計算機を使う新人研究者と年配の先輩研究者
  - お互いをライバル視し、反目している
  - 今までは、それぞれ独自に計算機を選定し、設置場所も別々
  - 研究者ごとにそれぞれのクラスタを占有して使っていた

- HPCシステムズの御提案
  - 全ての計算機を1ラック内に収め、LSFにより管理。同時に利用できる計算機リソースが単純計算で2倍に！
  - 計算機資源を公平に配分するフェアシェアの導入で不公平感を感じることなく計算機を使用可能！
  - 人間関係も改善しました

LSF活用事例::長いジョブと短いジョブ

- ブレーキ装置製造等で有名な機械メーカ様
  - 24時間近くかかる長いジョブ
  - 15分程度で終了する短いジョブ
  - これらが混在しているため、一旦長いジョブが走り出してしまうと、短いジョブをLSFで投入しても、業務時間中に終了しない程の待ちが発生
  - クラスタの利用効率が下がってしまう

- HPCシステムズの御提案
  - 3つのキューを作成しました
    - 平日9:00～18:00 30分で強制終了されるキュー[DAY]
    - 平日夜間18:00～9:00 6時間で強制終了されるキュー[NIGHT]
    - 休日 36時間で強制終了されるキュー[HOLIDAY]

LSF活用事例::ホワイトボードで資源管理？

- 某大学計算機センター様
  - 計算機の使用予定を台数分ホワイトボードとマグネットで管理
  - ホワイトボードの更新を忘れると、計算機の利用状況と乖離していく
  - 競合等が発生し、ジョブが異常終了することも

- HPCシステムズの御提案
  - LSFによる管理を御提案しました
  - LSFなら自動的に計算機の状況を収集・反映
  - 過去の稼動実績もトレースできます

ジョブ管理システムに LSF を選ぶ利点

- LSFについてのトラブル対応経験が豊富なため、殆どのトラブルは弊社のサポート窓口で解決可能
- 有償のジョブ管理システムなので、万一の際にもPlatform Computing 社の支援が受けられる
- 設定をお客様や技術者が作り込まなくても、初めから出来が良い

他のジョブ管理システムと LSF の違い

- LSFは他のジョブ管理システムと比較しても、圧倒的な使いやすさと機能を誇ります

	LSF Ver.7	PBS Pro 9.1	SGE 6.0	
使いやすさ/バイナリの直接投入	◎	×	×	PBS,SGEはシェルスクリプト作成によるジョブ投入のみサポート
並列ジョブの追跡	◎	○	×	LSFならOrphan process（孤児プロセス）の発生を確実に防止
Web GUI	◎	×	×	Platform Management Console(7.0以降), Web GUI(それ以前)
フローティングライセンス	◎	○	×	フローティングライセンスのサポート

※2009/09/09 弊社調べ

LSFで動作実績のあるアプリケーション

- 弊社では、次のようなアプリケーションに対して、LSFでの動作確認を行っております

分野	アプリケーション例
量子化学計算	Gaussian Molpro GAMESS VASP WIEN2k CASTEP CPMD Quantum Espresso
分子シミュレーション	Amber GROMACS LAMMPS
構造・流体計算	Nastran STAR-CD ABAQUS FLUENT LS-DYNA
数式・統計処理	MATLAB
気象	WRF MM5

※これらアプリケーションは代表例であり、これら以外にも設定なしで動作したり、仕組みを作ることで動作するアプリケーションも多数あります。

LSFの使い方:: JOBの投入方法 その1

- JOB の投入方法(基本編)

```
$ bsub ./a.out
```

```
Job <812> is submitted to default queue <normal>.
```

- 基本は、コマンドの先頭に bsub を付けるだけ！

LSFの使い方::投入したJOBの確認方法

- 当該ユーザで現在実行中のJOBを表示

```
$ bjobs
```

```
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
812     hpc     RUN   normal     hpcs01.hpc. 2*hpcs01.hp *pirun cpi Feb 26 18:21
                                  2*hpcs02.hpc.co.jp
```

- 全てのユーザで現在実行中のJOBを表示

```
$ bjobs -u all
```

- 当該ユーザで最近完了したJOBを含め、全ての状態のJOBを表示

```
$ bjobs –a
```

```
HOST_NAME          STATUS      JL/U    MAX  NJOBS   RUN  SSUSP  USUSP   RSV
hpcs01.hpc.co.jp   ok          -       2    0       0    0      0       0
hpcs02.hpc.co.jp   ok          -       2    0       0    0      0       0
hpcs03.hpc.co.jp   ok          -       2    0       0    0      0       0
```

- ホストの情報を表示

```
$ bhosts
```

LSFの使い方:: 並列JOBの投入方法

❖JOB の投入方法(MPICH編)

```
$ bsub -n 4 pam -g 1
  $LSF_BINDIR/mpichp4_wrapper $cwd/a.out
```

❖JOB の投入方法(OpenMPI編)

```
$ bsub -n 4 pam -g 1
  $LSF_BINDIR/openmpi_wrapper $cwd/a.out
```

※ -n の後の数字は並列数

※ “pam –g 1” を書く事でゾンビプロセスの残存を防止

LSFの使い方:: 並列JOBの投入方法2

- JOB の投入方法(Gaussian03 Linda なし, SMP 4 コア編)

```
$ bsub -o out -n 4 "g03 < test397.com > test397.log"
```

※但し、インプットファイル(この場合は test397.com)の中に 4コア SMP で流すオプションを書いておく必要があります。

例) %nproc=4

- JOB の投入方法(Gaussian03 Linda編, 4ノード, 各ノード8コア毎)

```
$ bsub -o out -n 32 lndjob "g03 < test397.com > test397.log"
```

※但し、インプットファイルの中に、4ノード並列、各8コアを使って流すオプションを書いておく必要があります。

例) %nproc=8
%nprocl=4

bsub コマンドのオプション

オプション	説明
-B	ジョブがディスパッチされると、電子メールを送信します
-H	ジョブが投入されると、ジョブを PSUSP 状態に保ちます
-I | -Ip | -Is	バッチ対話型ジョブを投入します。-Ip creates a pseudo-terminal. -Is はシェル モードに疑似端末を作成します。
-K	ジョブを投入し、ジョブの終了を待ちます
-N	ジョブが終了すると、電子メールでジョブ レポートが送信されます
-r	ジョブを再実行できるようにします
-x	排他実行（注意：キューの中に排他実行可能な設定がされている場合にこのオプションを用いて投入したJOBにのみ有効です。）
-b begin_time	指定の日時以降にジョブをディスパッチします。 日時の形式は [[month:]day:]hour.minute です。
-C core_limit	このジョブに属するすべてのプロセスに、プロセスごと(ソフト) のコア ファイル サイズ制限 (KB) を設定します
-c cpu_time[/host_name | / host_model]	ジョブが使用できる CPU 時間の合計を制限します。CPUの時間は [[month:]day:]hour:minute 形式で指定します。
-D data_limit	このジョブに属するすべてのプロセスに、プロセスごと(ソフト) のデータ セグメント サイズ制限 (KB) を設定します
-e error_file	ファイルに標準エラー出力を付け加えます
-ext[sched] "external_scheduler_options"	アプリケーション固有のジョブの外部スケジューリングオプションです (extsched オプションを短く -ext とすることも可能です)
-E "pre_exec_command [arguments ...]"	ジョブを実行する前に、指定された実行前コマンドを実行ホスト上で実行します
-f "local_file op [remote_file]" ...	ローカル (投入) ホストとリモート (実行) ホストの間でファイルをコピーします。op は >、<、<<、><、<>のいずれか 1 つです
-F file_limit	このジョブに属するすべてのプロセスに、プロセスごと(ソフト) のファイル サイズ制限 (KB) を設定します。
-G user_group	指定したユーザ グループとジョブを関連付けます
-g job_group_name	指定したジョブグループとジョブを関連付けます。
-i input_file | -is input_file	指定したファイルからジョブの標準入力を取得します
-J "job_name[index_list] %job_slot_limit"	指定した名前をジョブに割り当てます ジョブ配列のIndex_list はstart[- end[:step]] の形式となっていますが、ここでstart ならびに end、step は正の整数、%job_slot_limit はどの時点においても実行可能なジョブの最大数です。
-k "chkpnt_dir [chkpnt_period] [method=method_name]"	ジョブをチェックポイント可能にして、チェックポイント ディレクトリならびにチェックポイント期間 (分)、チェックポイント方法を指定します
-L login_shell	指定のログイン シェルを使用して、実行環境を初期化します
-m "host_name [@cluster_name] [+[pref_level]] | host_group[+[pref_level]] ..."	指定したホストの 1 つでジョブを実行します。ホストまたはホスト グループの名前の後にプラス (+) がある場合は優先を意味します。オプションで、正の整数で優先順位のレベルを指定します 大きな数値は、それらのホストの優先順位が高いことを示します。
-M mem_limit	メモリ制限 (KB) を設定します
-n min_proc[,max_proc]	並列ジョブの実行に必要なプロセッサの最小数と最大数を指定します
-o output_file	ファイルに標準出力を付け加えます
-P project_name	指定したプロジェクトにジョブを割り当てます
-p process_limit	ジョブ全体のプロセス数の制限を設定します
-q "queue_name ..."	指定したキューにジョブを投入します
-R "res_req"	ホストのリソース要件を指定します
sla service_class_name	ジョブが実行されるサービスクラスを指定します
-sp priority	ユーザ割り当てジョブの優先順位を指定し、ユーザがキュー内のジョブをソートできるようにします
-S stack_limit	このジョブに属するすべてのプロセスに、プロセスごと(ソフト) のスタック セグメント サイズ制限 (KB) を設定します
-T thread_limit	ジョブ全体の並行スレッド数の制限を設定します
-t term_time	ジョブの終了期限は、[[month:]day:]hour:minute 形式で指定します
-U reservation_ID	brsvadd で作成したアドバンス予約を使用します
-u mail_user	指定した電子メール アドレスにメールを送信します
-v swap_limit	ジョブ全体の合計プロセス仮想メモリ制限(KB) を設定します
-w 'dependency_expression'	依存式が TRUE と評価した場合にジョブを実行します
-wa '[signal | command | CHKPNT]'	ジョブ管理アクションが起こる前にジョブアクションがあるように指定します
-wt '[hours:]minutes'	ジョブ警告アクションを伴うジョブ管理アクションが発生する前に時間を指定します
-W run_time[/host_name | / host_model]	ジョブの実行時間制限は、[[month:]day:]hour:minute 形式で指定します
-Zs	lsb.params のJOB_SPOOL_DIR で指定したディレクトリにジョブのコマンドファイルをスプールします
-h	コマンドの使用法を stderr に出力して終了します
-V	LSF のリリース バージョンを stderr に出力して終了します

出典: LSFクイックリファレンス

クラスタ情報の表示

コマンド	説明
bhosts	ホストおよびホストの静的リソースと動的リソースを表示します
bhpart	ホスト パーティションの情報を表示します
bmgroup	ホスト グループの情報を表示します
bparams	調整可能なバッチ システム パラメータの情報を表示します
bqueues	バッチ キューの情報を表示します
brsvs	アドバンス予約を表示します
bugroup	ユーザ グループの情報を表示します
busers	ユーザとユーザ グループの情報を表示します
lshosts	ホストとその静的リソースの情報を表示します
lsid	現在の LSF バージョン番号ならびにクラスタ名、マスタ ホスト名を表示します
lsinfo	負荷共有設定情報を表示します
lsload	ホストの動的負荷インデックスを表示します

JOB のモニタ

コマンド	説明
bhist	ジョブに関する履歴情報を表示します
bjgroup	ジョブグループの情報を表示します
bjobs	ジョブに関する情報を表示します
blimits	リソース割り当て制限についての情報を表示します
bpeek	未完成ジョブの stdout (標準出力) および stderr (標準エラー) を表示します
bsla	目標志向のSLA スケジューリング用のサービスクラス設定についての情報を表示します
bstatus	外部ジョブ状態のメッセージとデータ ファイルの読取りまたは設定を行います

JOB の制御

コマンド	説明
bbot	キュー内の最後のジョブを基準にして、保留ジョブを移動します
bchkpnt	チェックポイント可能なジョブでチェックポイントを実行します
bgadd	ジョブグループを作成します
bgdel	ジョブグループを消去します
bkill	ジョブにシグナルを送信します
bmig	チェックポイントまたは再実行可能なジョブを移行します
bmod	ジョブ投入オプションを修正します
bpost	メッセージを送信し、データ ファイルをジョブに添付します
bread	ジョブからのメッセージと添付されたデータ ファイルを読み取ります
brequeue	ジョブを中止してキューに再登録します
brestart	チェックポイントされたジョブを再起動します
bresume	中断されていたジョブを再開します
bstop	ジョブを中断します
bswitch	未完成ジョブを 1 つのキューから別のキューに移動します
btop	キュー内の最初のジョブを基準にして保留ジョブを移動します

出典: LSFクイックリファレンス