CBPF-NT-002/99

LSF – Load Sharing Facility

por

Luis Gustavo S. M. Bessa
Marita Maestrelli

Centro Brasileiro de Pesquisas Físicas – CBPF
Rua Dr. Xavier Sigaud, 150
22290-180 – Rio de Janeiro – RJ - Brasil

PREFÁCIO:

Sendo o CBPF uma instituição de pesquisa na área de física é evidente que se tenha uma demanda muito grande de equipamentos computacionais aptos a realizar cálculos “pesados” no menor tempo possível.

Afim de atender essas necessidades o CBPF possui um parque de estações RISC, sejam elas SUN ou IBM. Porém, por mais poderosas que estas workstations possam ser, o CBPF, até o ano de 1998, não possuía nenhum equipamento de grande porte. Pensando nisso, foi comprado um computador muito poderoso. Este é o Ultra HPC4000 da Sun Microsystems.

O Ultra HPC4000 é um computador de processamento paralelo, com 10 processadores RISC de 250 MHz, 2.5 GB de memória RAM e uma velocidade de processamento muito alta, portanto, o mais poderoso a serviço do CBPF.

Este computador está ligado à rede SUN, e pode ser acessado remotamente pelo endereço **sol.cat.cbpf.br** com IP: 152.84.253.53.

Quando se trata de um servidor utilizado por vários usuários, rodando programas que exigem muito do processador, até mesmo uma máquina de grande porte como a SOL (como chamaremos a estação Ultra HPC4000) pode sofrer problemas de rendimento, se não for bem utilizada e gerenciada. Frente a esta circunstância, tornou-se necessária a utilização de um eficiente software de gerência chamado de LSF (Load Sharing Facility).

Nesta nota técnica serão apresentadas as vantagens de se usar o LSF para rodar programas mais “pesados”, tanto do ponto de vista do usuário como do gerente da máquina. Serão mostrados e explicados também todos os comandos necessários para que o usuário execute e até gerencie seus programas, de modo a obter o melhor possível do potencial da máquina. Esses comandos serão de valia também para qualquer outra maquina que tenha o LSF instalado(no nosso caso, somente a máquina **sol**).

ÍNDICE:

1. Introdução:

O software LSF não é muito conhecido pela maioria dos usuários de computador, porém sua utilização é de extrema facilidade.

O LSF, do ponto de vista do usuário, é uma interface entre ele e o computador que permite executar comandos, scripts ou programas, de maneira interativa, ou por meio de **batches**.

Do ponto de vista do administrador do sistema, o LSF é uma eficiente ferramenta de gerência dos processos que *rodam* na máquina. Portanto, quando se trata de um equipamento de grande porte e com alto índice de utilização, como a estação sol, para benefício tanto do usuário como do administrador é necessária a implementação do LSF. O usuário que adotar o LSF como instrumento de execução de processos, perceberá o aumento na velocidade de resposta que consequentemente ocorrerá no sistema.

Outra vantagem do LSF é a possibilidade de criação de **filas**, ou "*queues*", que é uma ferramenta de extrema valia para evitar a sobrecarga do sistema, e melhor aproveitá-lo em horários de pouco uso, como por exemplo, à noite.

Outra possibilidade de aplicação do LSF é na criação de **clusters**, que no caso da **sol**, será composto apenas por ela.

Por fim, mas também muito atrativa, está a opção de execução e monitoramento dos jobs ou processos, por meio de um interface gráfica, que é uma boa opção para os não acostumados com o sistema UNIX.

2. Conceitos:

Antes de iniciarmos na parte dos comandos, o leitor deve estar ciente de alguns conceitos como os de **batch**, **filas**, **processamento paralelo** e **cluster**.

2.1 Batch:

Quando executamos algum processo em um computador, seja ele um comando do shell, um script ou mesmo um programa, podemos faze-lo de 2 formas: interativamente e por batch.

Interativamente significa rodar o job em tempo real, ou seja, estar *logado* na maquina servidora, iniciar o processo, esperar logado toda a sua execução, e, por fim, ler a saída que o job nos gerou. É o que acontece quando digitamos o comando "*ls"* e observamos a lista de arquivos que nos é mostrada como saída. Este procedimento é ideal quando se trata de jobs com respostas rápidas (no máximo 3 minutos).

Quando se trata de jobs com maior tempo de duração é interessante rodá-los por meio de batches. Mas o que vem a ser um batch? Do Inglês, batch significa remessa, e é exatamente isso que o batch é. Quando rodamos um programa por meio de batch, nós não o executamos diretamente, mas sim enviamos ao computador um pedido de execução no qual estão todas as informações necessárias para o computador executá-lo. Portanto, sendo o envio quase instantâneo, não é necessário ficarmos observando o término do processo, e muito menos ficarmos logados. Após o job ser submetido, este ficará a cargo apenas da máquina que irá processá-lo, e, ao término, o resultado será retornado por meio de um e-mail, ou escrito em um arquivo. Assim, não importa se a execução do programa durará 1 hora ou 1 dia, para o usuário basta submeter a tarefa, e depois ler o resultado.

2.2 Filas:

O conceito de filas está relacionado ao de batches, pois quando submetemos um job por meio de batch, este não é executado instantaneamente ao chegar no computador. Na verdade ele entra em uma fila de execução atrás de outros batches previamente submetidos pelo mesmo ou por outros usuários. O job então pode ser mandado para a fila *default* se nada for especificado, ou pode ser enviado para uma fila especial. Essas filas especiais são definidas pelo gerente do sistema, e possuem características específicas, como por exemplo nível de prioridade. O nível de prioridade está ligado diretamente à "fatia" do processador que o job pode utilizar, portanto uma aplicação interessante foi a criação da fila *night* que caracteriza-se por executar os jobs apenas a noite, porém dando um maior nível de prioridade a esses jobs.

2.3. Processamento paralelo

Estamos acostumados com a multitarefa presente na maioria dos sistemas operacionais, porém o que vemos não é um processamento paralelo, e sim a divisão do processamento de um programa em pequenas partes que são processadas intercaladamente com partes de outro programa, dando a impressão de que 2 ou mais programas estão sendo executados ao mesmo tempo. Entretanto, só podemos ter o processamento de mais de um programa, ou parte dele, ao mesmo tempo, se tivermos mais de um processador trabalhando paralelamente.

Portanto, só teremos processamento paralelo quando tivermos um computador com mais de um processador, como é o caso do Ultra HPC4000, ou quando tivermos mais de um computador interligados em rede e controlados por um software específico.

2.4 Cluster:

O LSF gerencia um ou mais clusters. Um cluster é um conjunto de computadores ligados em rede, e gerenciados por um software. No caso do Ultra HPC4000, só ele pertence ao cluster, porém poderiam haver outros computadores, desde que estivessem ligados em rede. Esses softwares de gerenciamento tratam o cluster como se fosse um único computador com um ou mais processadores, mas com memória distribuída.

Por exemplo, se dois computadores formam um cluster gerenciado pelo LSF, ao enviarmos um job, ele, a não ser que especifiquemos, irá para o cluster e não para um computador específico. Entretanto o LSF se encarregará de executar o job no computador mais adequado. Outra vantagem é a possibilidade de se executar o job paralelamente nos 2 computadores do cluster.

3. Comandos do LSF:

Todos os comandos estão em : /usr/local/lsf

3.1 Informações sobre o sistema:

- **lsid** dá informações a respeito do nome do cluster, e do master (computador que gerencia o cluster), locais. Pode ser usado também o comando **lsclusters**.

Ex:
sol[gustavo]% lsid

My cluster name is sol_cl
My master name is sol

- **lshosts** da informações a respeito da configuração e recursos de todos os hosts do cluster.

Ex:

sol[gustavo]% lshosts

HOST_NAME	type	model	cpuf	ncpus	maxmem	maxswp	server	RESOURCES
sol	DEFAULT	DEFAULT	1.0	10	2560M	3017M	Yes	( )

- **bhosts** mostra informações sobre a situação dos batches no(s) host(s).

Ex:

sol[gustavo]% bhosts

HOST_NAME	STATUS	JL/U	MAX	NJOBS	RUN	SSUSP	USUSP
sol	ok	-	-	0	0	0	0

-bqueues mostra as características das filas, e a atual situação de uso das mesmas, como processos esperando ou sendo executados. Observe:

Ex.:

Sol[gustavo]% bqueues

QUEUE_NAME	PRIO	NICE	STATUS	MAX	JL/U	JL/P	NJOBS	PEND	RUN	SUSP
priority	43	10	Open:Active	-	-	-	0	0	0	0
profs	43	10	Open:Active	-	-	-	0	0	0	0
fast	43	10	Open:Active	-	-	-	0	0	0	0
night	40	10	Open:Active	-	-	-	2	2	0	0
general	1	0	Open:Active	-	-	-	0	0	0	0

A partir daí podemos ver quais são as filas, se elas estão ativas (disponíveis), qual os respectivos níveis de prioridade de envio e de processamento, o limite máximo de processos em cada fila (MAX), o número de jobs que estão atualmente nas filas e a situação dos mesmos (espera, em execução e suspenso).

- bjobs mostra uma lista com os jobs que estão na fila ou já sendo executados. Se for utilizada a extensão " –u all ", serão listados os jobs de todos os usuários. Observe:

Ex:

sol[gustavo]% bjobs –u all

JOBID	USER	STAT	QUEU	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1004	gustavo	RUN	fast	sol	sol	math2	12:45
1235	mario	RUN	fast	sol	sol	phisics	12:44
1234	sergio	SSUSP	profs	sol	sol	rmnexp	11:59
1250	carina	PEND	profs	sol	sol	highener4	12:05

Acima, pode-se observar que o comando gerou uma lista contendo informações tais como o número de identificação do job, o usuário que enviou o job, o status desse job (nesse caso, RUN quer dizer que o job está em estado de execução, SSUSP significa que o job foi suspenso e PEND significa que o job ainda esta na fila), a fila para qual ele foi enviado, de que host ele foi enviado, em que host ele esta sendo ou será executado, o nome que foi dado a ele e o horário de envio do mesmo.

3.2. Submetendo jobs:

- **bsub** é o comando utilizado para enviar os jobs a serem executados. Como já foi mencionado, esses jobs podem ser comandos de unix, scrips ou programas executáveis.

O comando bsub possui alguns parâmetros importante. Observe:

➔ -q <fila desejada>

Ao enviarmos um job, devemos especificar para que fila ele irá. Caso contrário, o job irá para a fila "default".
Ex:

sol[gustavo]% bsub –q fast ls

Job <1525> is submitted to queue <fast>.

No exemplo acima o usuário "gustavo", através do comando bsub, enviou o comando "ls" para a fila "fast". Logo após, o lsf da uma resposta. Nessa resposta vem o número do job e para qual fila ele foi submetido.

➔ -n <processors>

Quando executamos um job em uma máquina de mais de um processador, ou em um cluster com mais de um computador, podemos rodar o job paralelamente usando a extensão –n. Lembrando-se que nem sempre é vantajoso rojar um job em paralelo, principalmente no que se trata de jobs rápidos, pois há perda de velocidade na comunicação entre os "sub-jobs" (ao rodar um job em x processadores, serão criados x sub-jobs diferentes) e, para o job ir da fila de espera para a execução, os x processadores deverão ficar disponíveis.

Ex:

sol[gustavo]% bsub – n 4 prog1

Ou seja, o programa prog1 foi enviado para ser executado em 4 processadores.

➔ -o <arquivo>

A extensão –o redireciona a saída da execução do job para um arquivo qualquer, ao invés do default que é retornar via e-mail.

Ex:

sol[gustavo]% bsub -o saida prog1

O programa prog1, agora terá sua saída redirecionada para o arquivo saida.

➔ -i <arquivo>

Funciona semelhantemente ao –o, porém ao invés de especificar o arquivo para onde irão os dados de saída, especifica o arquivo de onde virão os dados de entrada do programa.

➔ -u <e-mail>

Esta serve para redirecionar o endereço de e-mail que irá receber a saída da execução do job.

Estas são as principais extensões utilizadas no envio de jobs, lembrando que as mesmas podem ser combinadas.

Ex:

sol[gustavo]% bsub -q fast –i entrada -o saída –n 4 prog1

3.3. Manipulando os jobs:

- bkill é utilizado para finalizar um job que está na fila de espera, ou em execução. Através do comando **bjobs**, vê-se o número do job (JOBID). Lembrando que esse comando apaga a job.

Ex: sol[gustavo]% bkill 1004

Job <1004> has being terminated

- bstop é utilizado para deixar o job em estado SSUP (suspenso). Ou seja, ele fica parado, tanto se estiver na fila como se estiver em execução. O conveniente é que ele pode se retomado exatamente de onde parou através do comando **bresume**. A linha de comando para o **bstop** e o **bresume** é semelhante ao **bkill**.

3.4. Execução interativa de jobs:

- lsrun é o comando utilizado para executar os jobs interativamente. O lsrun funciona da mesma maneira que a execução direta do job, ou seja escrevendo direto o comando ou o nome do arquivo executável. A diferença é que este comando chama o LSF, e assim, pode se fazer uma melhor administração do sistema, como controle de carga e estatísticas de uso. Este comando é indicado para execução de jobs rápidos, que necessitem de rapidez na visualização dos resultados, portanto não é recomendado para jobs mais pesados.

Ex:

sol[gustavo]% lsrun ls

Este é um comando tão simples que não necessitaria ser monitorado pelo LSF, porém nota-se que a saída foi igual a que seria se fosse digitado apenas ls, a diferença está no fato de que o LSF monitorou a execução:

/opt/SUNWlsf/sun/sparc# ls
bin etc lib
/opt/SUNWlsf/sun/sparc# lsrun ls
bin etc lib

4. Interfaces Gráficas:

Todos os comandos estão em : /usr/local/lsf

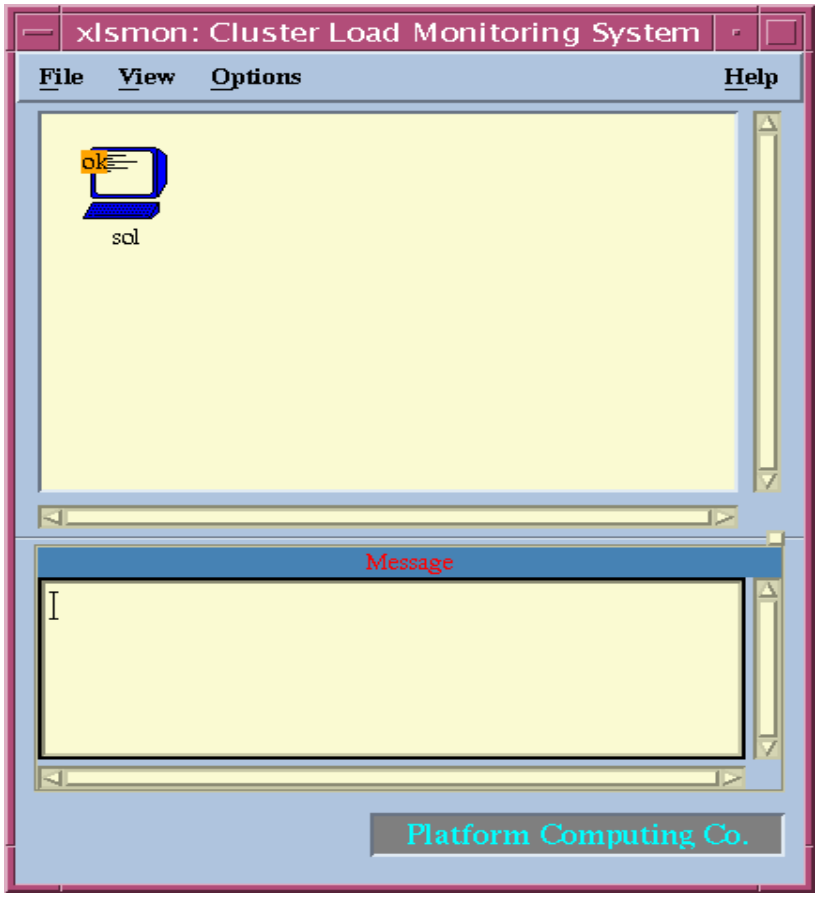
Todos os comandos acima descritos podem ser executados de maneira mais amigável fazendo uso de três programas gráficos: o **xlsmon**, o **xlsbatch** e o **xlsadmin**. Porém os comandos foram mostrados devido ao fato de muitos usuários preferirem à interface gráfica, pois consegue-se maior agilidade, uma vez que janelas gráficas são mas pesadas para serem abertas, e se o usuário não tem uma boa velocidade na conexão de rede com o computador em que o LSF reside, a abertura de uma janela gráfica se torna impossível. Programas de telnet em PC's também não permitem a abertura de janelas gráficas. Portanto é recomendado que o usuário aprenda pelo menos alguns comandos mais básicos vistos anteriomente. A seguir seão mostrados os três programas gráficos:

4.1 xlsmon:

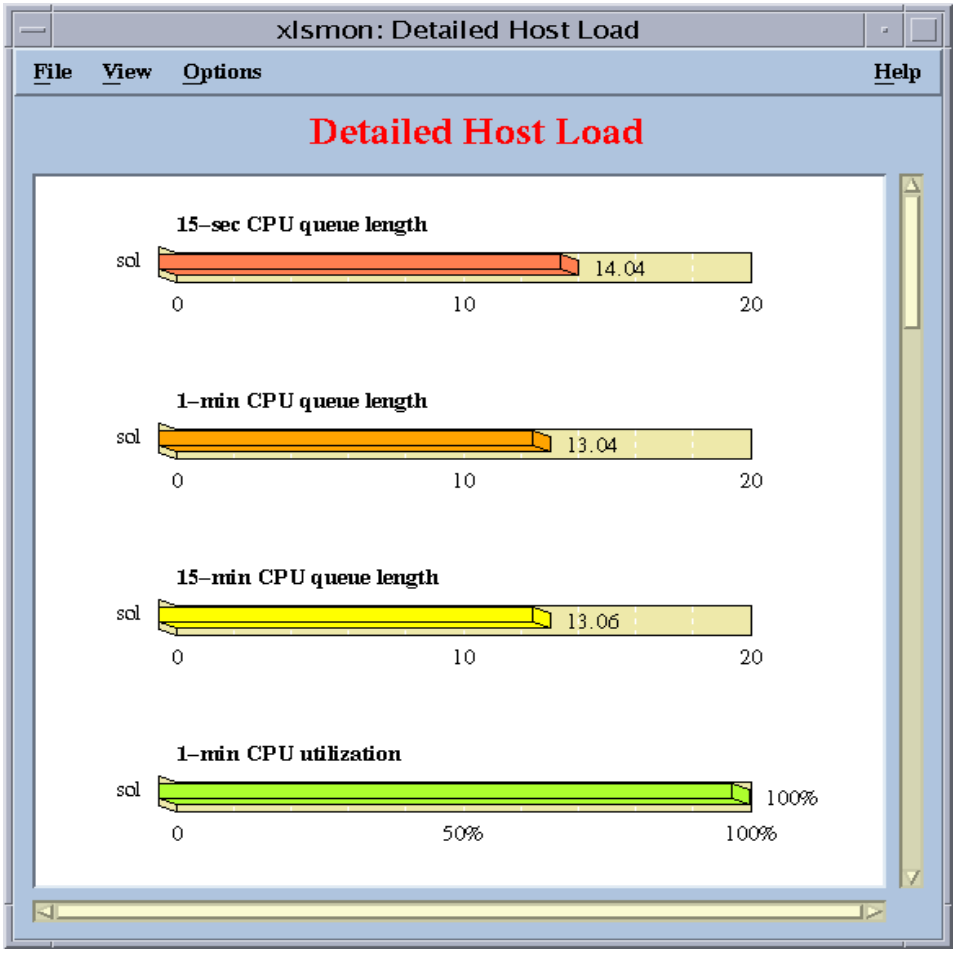
Executa-se o **xlsmon** do seguinte modo:

sol[gustavo]% xlsmon

E a seguinte tela gráfica aparecerá:



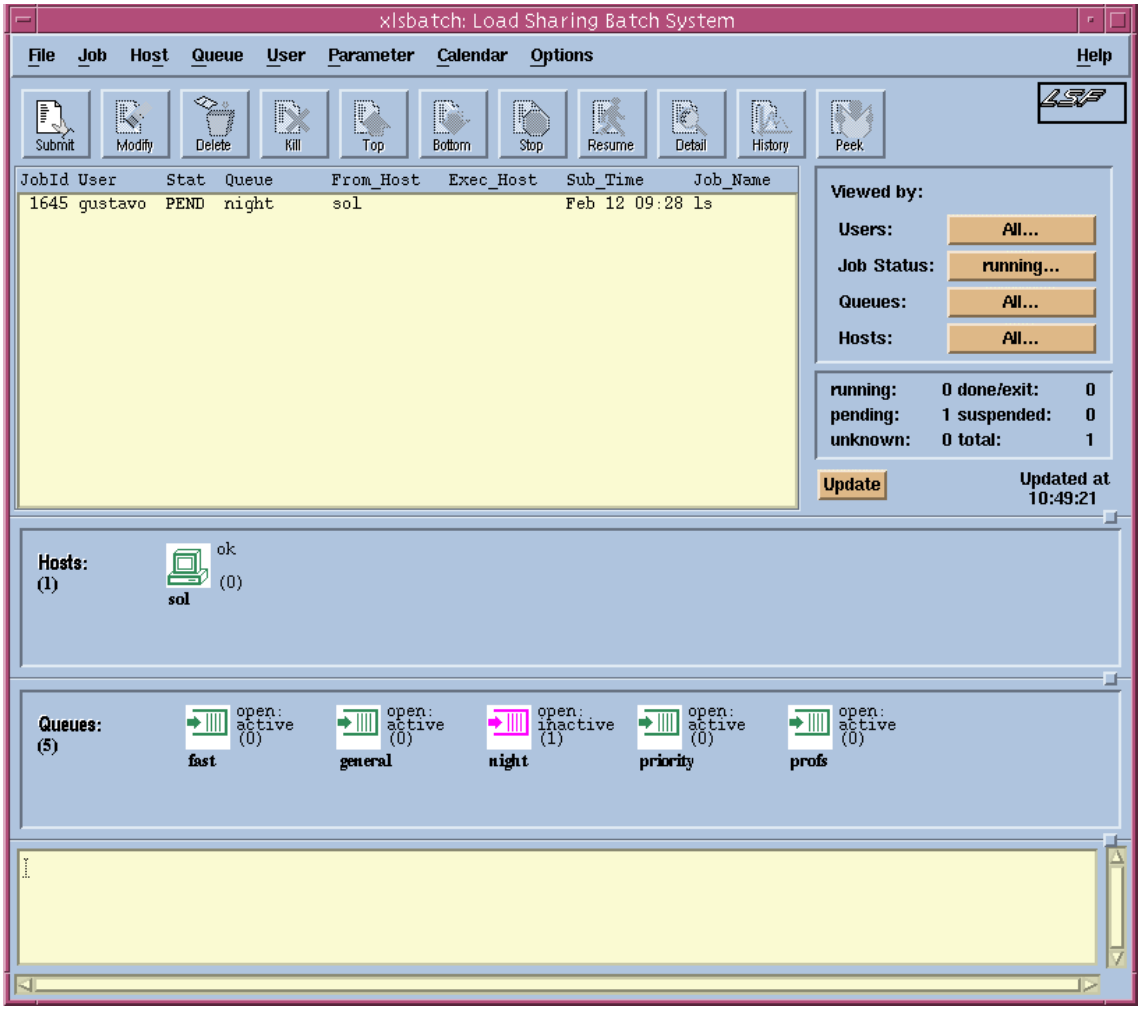
Nesta janela poderão aparecer mensagens do administrador, assim como várias informações sobre o cluster. Clicando em **view** e em seguida em **detailed load**, veremos:



De onde podemos ver, graficamente, as informações mais importantes sobre o sistema.

4.2 xlsbatch:

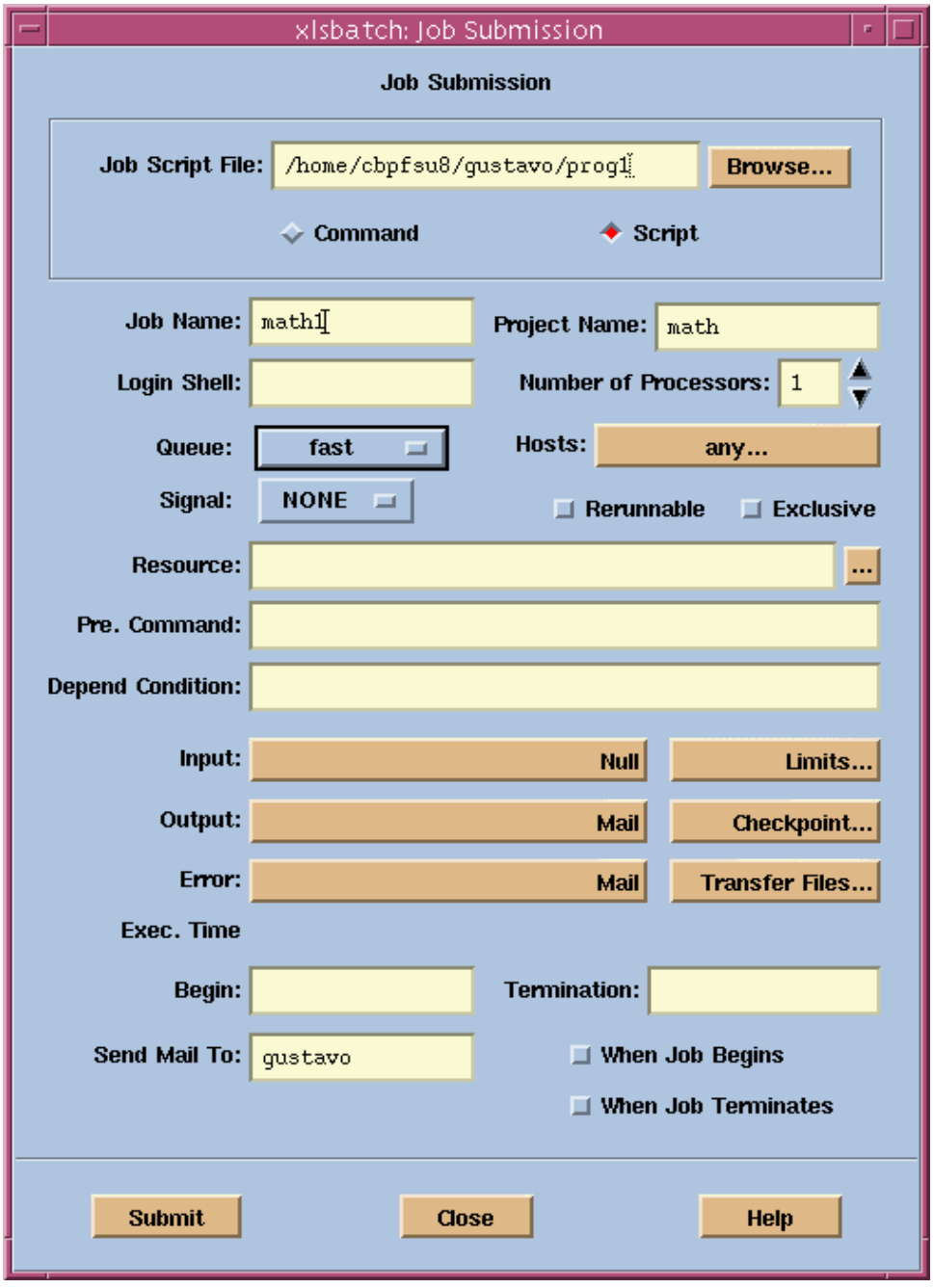
Ao digitarmos **xlsbatch**, aparecerá a seguinte janela gráfica:



Através dessa janela, temos o controle total sobre os nossos jobs. Por exemplo:

- clicando no ícone **submit** pode-se enviar os jobs (será melhor explicado adiante);
- clicando no ícone **modify** pode-se fazer alterações nos jobs que estão sendo executados ou estão na fila (deve-se primeiro selecionar o job desejado);
- pode-se finalizar o job;
- paralisar e reiniciar o job;
- analisar o *status* de todos os jobs que estão na fila ou em execução;
- visualizar quais são as filas e as características das mesmas;

Ao clicarmos no icone onde está escrito **submit**, aparece a seguinte janela:



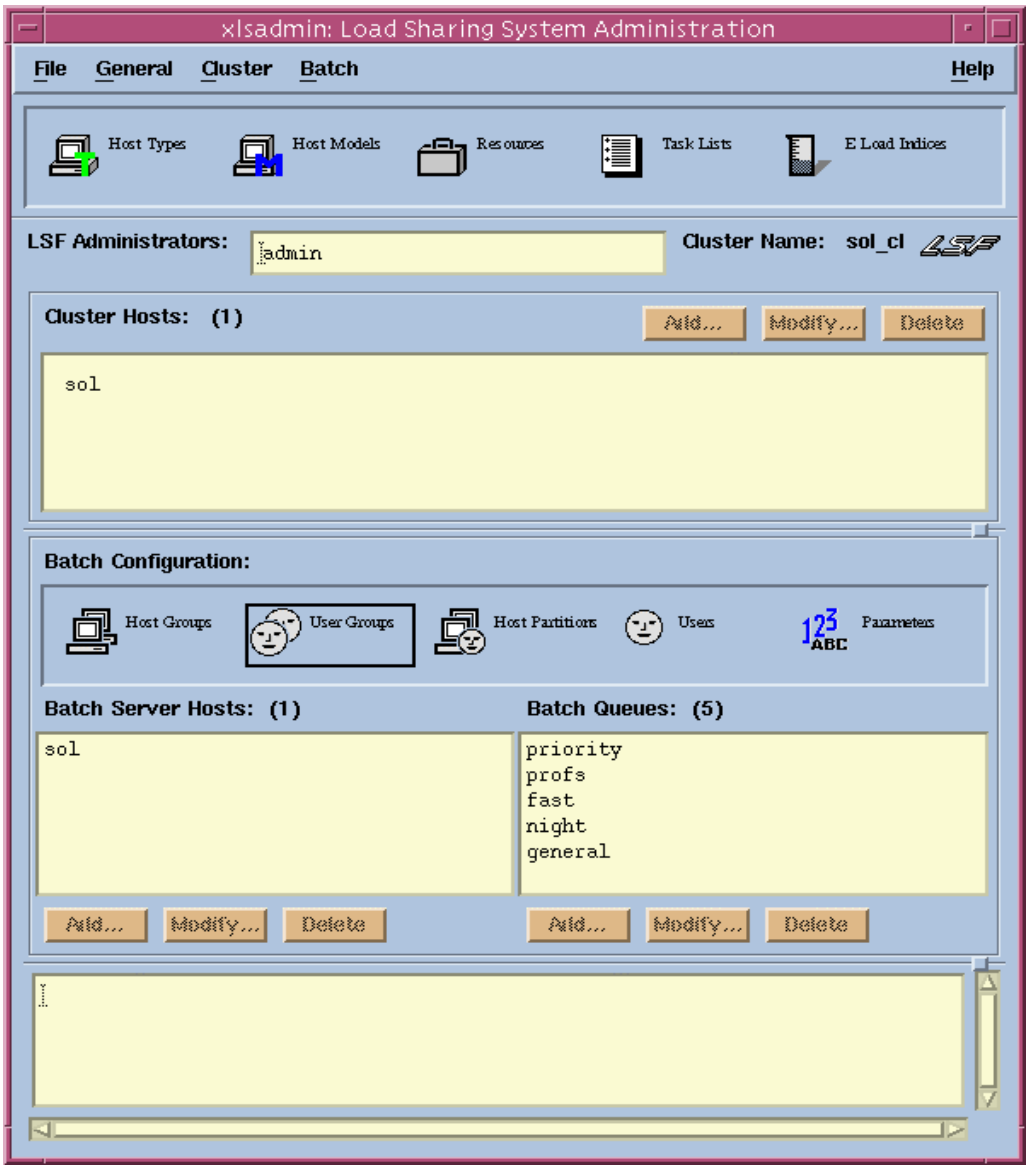
Através dessa janela, podemos, facilmente configurar todas as opções possíveis no envio de um job. Observe as mais importantes:

- a primeira opção é que job será enviado (podendo ser um comando ou um script);
- pode-se nomear o job e o projeto a qual ele pertence;
- o número de processadores a serem utilizados;
- a fila para qual ele será enviado;
- o arquivo com os dados de entrada (caso exista);
- a saída do resultado (arquivo ou mail);
- o e-mail de destino do resultado do job executado (caso a saída seja por e-mail);

Como pode ser observado existem muitas outras opções, porém as listadas acima são as mais importantes.

4.3 xlsadmin:

Digitando **xlsadmin**, temos:



O xlsadmin é uma ferramenta de uso do administrador do sistema, embora o usuário possa executá-la, este não poderá fazer nenhuma alteração. Para o administrador, a utilização desta janela é extremamente importante. Através dela pode-se configurar todas as opções do sistema, e como ela só serve para isso, fica ao usuário apenas a nível de curiosidade.

5. Conclusão:

A utilização do LSF é interessante tanto para o administrador da ***sol*** como para os seus usuários. A seguir será mostrada a atual situação das filas que estão em operação na estação ***sol***.

5.1. Definição das Filas:

- **Priority** (fila default do LSF, para onde o job vai caso de nenhuma fila seja especificada – baixa prioridade);
- **Profs** (fila para os participantes do projeto FINEP de física teórica - alta prioridade);
- **Fast** (fila para jobs rápidos – alta prioridade);
- **Night** (fila para execução de jobs durante a noite – alta prioridade);
- **General** (fila para uso geral, sem restrições de horários ou tempo de duração – média prioridade);

As filas **priority** , **fast** e **genera**l podem ser usadas por qualquer usuário;
As filas **profs** e **night** somente por usuários do grupo de física teórica

5.1. Utilizando o LSF:

Resumindo:

- O LSF (load sharing facility) é uma interface entre o usuário e o computador que permite executar aplicações paralelas ou sequenciais de maneira interativa ou por meio de *batches*.
- A estação da SUN habilitada para seu uso é a ***sol*** , que tem a seguinte configuração: SUN Ultra HPC 4000 Server
  Processadores: 10 de 250 MHz cada
  Memória: 2,5 GB
  Cache externo: 4 MB
  Unidade de fita DAT (1/4 ")
  CD-ROM 12 X
  Nome/IP: sol.cat.cbpf.br / 152.84.253.53
- Localização dos comandos e das *manpages* do LSF:
  /usr/local/lsf comandos
  /usr/local/lsfman manpages(usadas com o comando **man**)
- Aplicação das filas :
  Exemplos:
  sol [tsallis]% bsub –q profs kaos2000

Usuário do grupo de física teórica **tsallis** , colocando para executar o programa **kaos2000** na fila de prioridade alta **profs**

sol [guest]% bsub –q fast rees

Usuário sem privilégios **guest** que coloca o programa **rees** para executar na fila **fast** de prioridade alta (mas para jobs com no máximo 5 minutos de tempo de execução).

sol [guest]% bsub rees

Agora o usuário **guest** coloca para rodar o programa **rees** na fila **default** (priority) de baixa prioridade, mas sem limite de tempo.

sol [guest]% bsub –q general rees

O usuário **guest** coloca para rodar o mesmo programa **rees**, mas desta vez utilizando a fila **general**. Só que desta vez o tempo do programa na máquina pode ser maior, pois a fila **genera**l embora tenha maior prioridade que a fila **priority**(a fila *default*), ela tem um limite menor de jobs executando.

- Utilização dos comandos: **batch** e **at**
  Os comandos **batch** e **at** executarão sempre a fila *default* do LSF , isto é, os programas rodados através desses comandos rodarão em baixa prioridade.
  Sendo assim, utilize sempre o pacote LSF para manipulação de seus programas, pois poderá acompanhá-los(monitorá-los), durante a execução, mais facilmente.

6. Bibliografia:

- Manuais do LSF;
- Sites na Internet;

- http://www.platform.com (site do fabricante)
- http://wwwinfo.cern.ch/pdp/lsf/ (muito bom)
- http://www.science-computing.de/produkte/lsf-en.html
- http://www.glue.umd.edu/htdocs/lsf-docs/