Acesso

O que acontece com o job caso o certificado expire antes do final da execução?

Caso o proxy expire, o sistema não vai conseguir atualizar a informação do andamento do job, nem copiar os arquivos de saída.
O job entrará em H (held), todavia ele não estará perdido.
Nesse caso deve-se abrir um novo proxy (myproxy-logon) e liberar o job (condor_release JOBID).

Preventivamente é possível solicitar um proxy maior que o padrão (12 horas) (com o parâmetro -t <horas>).

Sou coordenador de grupo de pesquisa, como faço para dar acesso ao GridUNESP a um novo membro do grupo?

O primeiro passo é cadastrar o projeto, caso ele ainda não exista. Para tal deve ser preenchido o Formulário de Submissão de Projeto Científico e enviado para o e-mail grid [at] ncc.unesp.br.

Uma vez que o projeto tenha sido cadastrado, então os usuários poderão utilizar o formulário de registro para requerer acesso ao GridUNESP. O acesso do usuário só estará liberado depois que o coordenador do projeto confirmar que aquele usuário pertence ao projeto em questão.

Esqueci minha senha de acesso ao !GridUNESP, como devo proceder para recuperá-la?

Deve ser enviado um email para a lista de suporte: support [at] ncc.unesp.br. Assim que a requisição for recebida pela equipe de suporte, uma nova senha é gerada e enviada diretamente para o email do solicitante. A senha é temporária, e deve ser alterada logo no primeiro acesso. O procedimento para troca de senha pode ser encontrado em: [[TrocarSenha]].

É possível acessar o GridUNESP do exterior (fora do Brasil)?

Nos não limitamos os acessos. Você pode acessar o GridUNESP de qualquer lugar.

Compilação de Jobs.

Preciso compilar meu job novamente na máquina do grid, ou apenas copiar minha versão já compilada?

Você não precisa recompilar se esse programa for da mesma arquitetura (linux 64bit - amd64 ou em64t) e ele não depender de bibliotecas dinâmicas "incomuns" no sistema.

Uma sugestão é a utilização do comando ldd, que pode ser utilizado para verificar as bibliotecas utilizadas pelo job compilado.

Submissão de Jobs

Eu preciso usar o mpirun.lam para rodar meus jobs. Antes de submeter devo executar o lamboot para requisitar os processadores. Porém, nem o mpirun.lam nem o lam-runtime estão instalados.

O GridUNESP não utiliza o lam-mpi e sim o Intel mpi. Se você submeter o binário diretamente, não precisa se preocupar em chamar qualquer programa mpirun, mpiexec ou lamboot, o sistema faz isso automaticamente para você.

O que você talvez precise é recompilar o seu programa para ele linkar com as bibliotecas corretas. No servidor Access existes essas ferramentas.

Estrutura do !GridUNESP

Dentro do cluster cada nó tem um disco?

Sim, tem um disco para scratch e arquivos temporários. Ele está montado em /tmp e possui 73GB.

Cada nó consiste de quantos processadores e quantos núcleos cada um?

Cada nó tem 2 processadores de 4 cores, um total de 8 cores por nó.

Preciso usar o pacote Gromacs em meus experimentos. Esse pacote está instalado?

Sim, temos o Gromacs compilado para precisão simples e dupla (_d), serial e MPI (_mpi).

Qual o escalonador do cluster e como funcionam as filas?

Grids computacionais, de forma pouco elaborada, são uma coleção de recursos, incluindo clusters. Dessa forma, ao utilizar um grid, você está utilizando clusters, mas não está interagindo diretamente com eles. Isso é feito através de um middleware.

Entrando em detalhes, o nosso cluster utiliza como escalonador o Torque + MAUI, que é uma versão livre do PBS. Assim, dentro do nosso cluster, temos o ambiente que você está acostumado, com qsub, qstat, pbsnodes...

Porém, você como usuário, não acessa isso diretamente, apenas através do middleware de grid, que utiliza a "linguagem" Globus ([[http://www.globus.org/]]). O condor que você tem acesso na access é uma forma de interface de usuário. Você poderia escrever arquivos de submissão em xml e enviar diretamente, mas isso é muito pouco prático.

O nosso cluster implementa 3 filas, limitadas por tempo máximo de excução: short (até 3h), medium (até 24h) e long (até 30 dias). Em qualquer uma você pode alocar processos paralelos.

O tamanho dos arquivos temporários, gerado pelo meu job, varia de cálculo para cálculo. Não é para passar muito de 8 GB. O que vai acontecer se encher o disco? O processo é cancelado automaticamente?

Não podemos afirmar com certeza o que acontecerá, depende do programa em questão. Acreditamos que o ponto chave é: o programa trata esse tipo de problema? Caso a resposta seja negativa, ele simplesmente dará uma falha de segmentação e o shell script copiará os dados parciais (se houverem) e apagará o resto.

Se o processo precisar de mais de 8GB por rodada, faltará disco. Será necessário especificar no seu arquivo de submissão quanto de espaço você precisa ou, ao invés de utilizar o /tmp, mudar o seu script para escrever no /store. A segunda solução é a que consideramos ideal para casos grandes do GAMESS, que gera arquivos de 60GB, 80GB cada rodada.

O /store é um file system compartilhado. Se apenas um cliente escrever nele, ele será mais rápido que o /tmp (~ 10x). Porém com escritas paralelas (de vários nós) ele tende a ficar mais lento. Porém acreditamos ser uma alternativa razoável.

Jobs Bloqueados

Estou tentando submeter meus jobs no Grid. Eles começam a rodar e após aproximadamente 10 minutos eles caem e o arquivo de log registra a mensagem apresentada mais abaixo. Como posso resolver isso?

```
012 (2158.000.000) 09/16 15:55:13 Job was held.
Globus error 155: the job manager could not stage out a file
Code 2 Subcode 155
```

O problema é que a execução está falhando, por qualquer razão, por isso os arquivos de saída não estão sendo criados.

Como eles não foram criados, o sistema de transferência do grid não consegue copiá-los, resultando no erro apresentado ao usuário.

A sugestão é rodar uma primeira vez sem especificar nenhum arquivo de saída (comentar a linha transfer_output_files). Depois da execução, será possível verificar no output e no error a saída do programa, permitindo o diagnóstico do erro.

Medição de Tempo de Execução

A fila de jobs é quase sempre formada por um único usuário, que chega a ter milhares submetidos. Sempre que submeto um job, ele entra na fila atrás desses milhares. Pergunto quanto tempo o meu job ficará aguardando nessa fila, antes de começar a rodar?

A fila que você observa na access (com o comando condor_q) não é uma "fila" entre usuários. Explicando melhor, ela só mostra uma fila intermediária de submissão, onde o único limite é a submissão de até 512 processos **por usuário**.

Assim, essa fila não tem nenhuma correspondência com a fila de execução do grid. Assim que o seu job é enviado, ele é imediatamente encaminhado para o grid (caso você tenha menos de 100 processos).

No cluster principal existe uma fila, mas que privilegia usuários do GridUNESP e usuários que utilizaram pouco recurso até o momento (um "fair share" com memória de 4 semanas).

É possível que o seu job esteja em IDLE por alguma outra razão.

Acabei de ver que no e-mail enviado pelo sistema (ou no .go.log) consta o tempo real (wallclock). Gostaria de saber o tempo de CPU.

O email mostra como o tempo gasto de execução zerado pois ele lista o quanto de recurso ele utilizou na máquina access, não no cluster. Sabe-se que essa informação é de pouco interesse ao usuário.

Caso você tenha utilizado o sistema de submissão simplificado (GO), os tempos de início e fim da simulação estão no arquivo .go.out.

Outras Questões