

Comparação entre os sistemas de cluster da MySQL AB e Emic Networks

Unidos venceremos

Já há muito tempo o MySQL é usado em aplicações críticas de empresas. Agora os primeiros sistemas de cluster para esse banco de dados já estão prontos para entrar de cabeça no mercado.

POR MICHAEL SPINDLER

Seja uma plataforma Web, um sistema de gerenciamento financeiro ou de comércio eletrônico, o centro de qualquer dessas aplicações é sempre um banco de dados. Se ele não funcionar bem, departamentos inteiros estarão condenados à estagnação. Daí a importância da prevenção e da palavra-chave: *alta disponibilidade*. Os produtos de código aberto, que aparecem cada vez mais em ambientes corporativos, precisam encarar a exigência da segurança contra falhas.

Este artigo apresenta dois sistemas de *clustering* para o banco de dados MySQL. Um deles é chamado simplesmente de *Cluster* e foi desenvolvido pelos mesmos desenvolvedores do MySQL, a empresa sueca MySQL AB. O outro tem nome semelhante: *Emic Application Cluster for MySQL* (ou, simplesmente, *EAC for MySQL*), da finlandesa

Emic Networks. Ambos vão além das atuais replicações mestre-escravo e são independentes de como o aplicativo foi desenvolvido. Ambos usam uma tecnologia chamada *Shared-Nothing*, com a qual cada nó dispõe de uma cópia própria do conteúdo de dados, que ele sincroniza com outros nós.

Aquisição de um cluster

O Cluster, da MySQL AB, não é tão novo quanto parece. Seu desenvolvimento iniciou-se na empresa Alzato, criada pela Ericsson, que em 2000 já trabalhava em uma solução de cluster para bancos de dados. Em 2003 a MySQL AB assumiu o controle da Alzato com o intuito de integrar o mecanismo de memória NDB (*Network Database*) ao servidor MySQL.

O MySQL funciona apenas como *front-end* para o Cluster NDB (figura 1).

O modelo de licenciamento produto não foi modificado

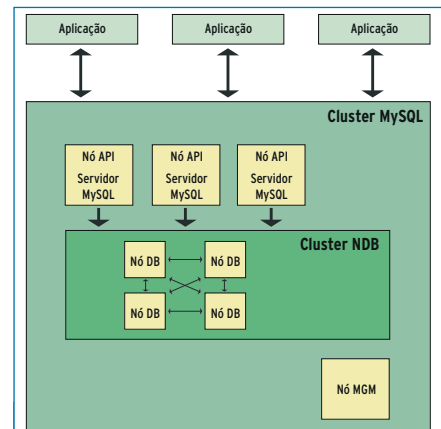


Figura 1: Os aplicativos que usam o Cluster consultam o servidor do MySQL. A administração dos dados propriamente dita é feita pelos nós do banco de dados do Cluster NDB, que serve de base para o MySQL.

por esse motivo, sendo que a tecnologia também está disponível nesse sistema de dupla licença.

Até pouco tempo só era possível uma réplica assíncrona entre os servidores MySQL; agora o Cluster realiza um ajuste sincronizado. Por enquanto ele ainda trabalha apenas como um banco de dados tipo *In RAM*: as informações são gravadas na memória para, só mais tarde, serem gravadas permanente no disco rígido durante um desligamento controlado. Além disso, durante a operação as modificações nos dados são enviadas regularmente para os discos, de modo que, mesmo que ocorra uma falha total de todos os nós, nem todos os dados são perdidos.



Nós API – o ponto fraco

O MySQL Cluster distingue três tipos de nós: o nó MGM (*Management Server*), o nó *Database* e o nó API. Define-se um nó como um programa em execução, e não um computador.

Para cada cluster deve haver pelo menos um nó MGM. Este é necessário para iniciar o cluster e também para administrar a configuração central da rede. Para a operação, o Management Server não é mais necessário. Somente quando os nós atualizarem suas configurações ele será consultado novamente.

Todos os nós juntos formam um grande banco. Cada nó tem seu próprio processo *nbd*, responsável pela manutenção dos dados. Os nós são totalmente independentes da interface de apresentação dos dados e, no cluster, servem como *backend*. Os servidores MySQL, com os quais os aplicativos se conectam, trabalham nos nós API. Enquanto que para o nó de bancos de dados foi implementado um gerenciador de falhas (*failover*) automático, a segurança dos nós API deve ser realizada pelo próprio administrador. O aplicativo registra uma falha em um nó API como uma falha total do Cluster, já que as conexões não são desviadas automaticamente para um outro nó API.

Proteção contra falhas no hardware

Os novos mecanismos de memória NDB podem ser usados junto com os formatos atuais de tabela, como o *MyISAM* e o *InnoDB*. Para criar uma tabela em formato NDB basta o usuário informar a opção `ENGINE NDBCLUSTER`; todo o resto acontece automaticamente. Uma tabela de um formato não-cluster vai para o disco rígido do nó API com o qual estiver conectado no momento.

O administrador pode dividir (particionar) completamente todos os dados do cluster. Assim, pode ocorrer de um determinado dado aparecer várias vezes no

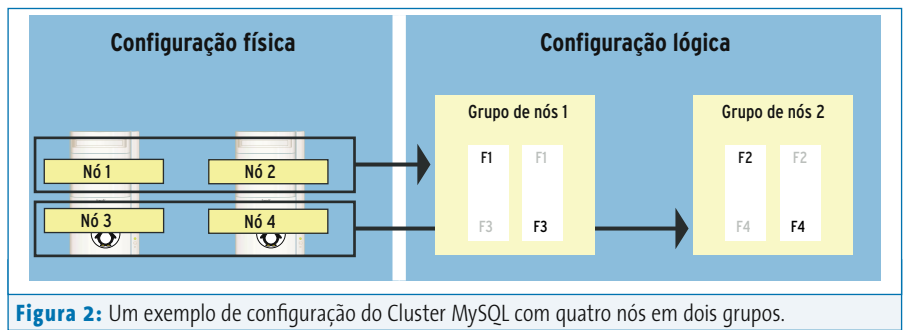


Figura 2: Um exemplo de configuração do Cluster MySQL com quatro nós em dois grupos.

cluster, mas ele não precisa ser salvo por cada nó. Para isso a configuração prevê o parâmetro `NoOfReplicas`, que informa quantas vezes a informação deve existir. O mínimo é uma, o máximo são quatro duplicatas. Com duas ou mais cópias os dados podem ser divididos entre vários computadores. Dessa maneira o cluster também protege os seus dados contra falhas de hardware.

Duplo é mais forte

No Cluster NDB há diferenciação entre os nós e grupos de nós. Os nós são unidades lógicas que salvam diferentes fragmentos de dados. Já os grupos de nós disponibilizam os fragmentos de dados anteriormente gravados. A figura 2 mostra uma configuração típica do Cluster NDB com 4 nós lógicos, dois grupos de nós e dois servidores físicos. Cada dois nós pertencem a um grupo de nós. No exemplo estão configuradas duas réplicas, o que significa que todos os dados são gravados duas vezes.

F1 a F4 nomeiam os fragmentos das áreas dos dados. Cada um dos nós salva dois fragmentos. Com essa divisão o Cluster continua funcionando, mesmo com falha de um dos servidores.

EAC for MySQL

A *Emic Networks* foi criada em 2000 e lançou no mercado a primeira versão do *EAC for MySQL* em outubro de 2002. Em janeiro foi lançado o cluster para o Apache, que é baseado na mesma tecnologia. O EAC para o MySQL é um produto

comercial. O software pode ser baixado a qualquer hora da página da web da EAC [1], mas para funcionar é necessário uma senha válida de licença. Os interessados recebem uma senha com validade limitada para teste. A licença EMIC abrange somente o software do cluster. O servidor MySQL modificado que vem junto tem que ser licenciado separadamente.

O EAC é formado por vários componentes, um módulo para o kernel, o software de replicação e de *Load Balancing* (balanceamento de carga) e o servidor MySQL modificado. O módulo do kernel é necessário para a filtragem dos pacotes e é utilizado através do Load Balancer. Se detectar que um dos servidores de dados está sobrecarregado ou sem comunicação, ele ajustará filtros de pacotes que desviam o tráfego de dados. Para a instalação do módulo os fontes do kernel devem estar disponíveis.

MySQL modificado

A versão EAC do servidor MySQL, que deve ser instalada em todos os nós, ainda é baseada no MySQL 4.0.14, ou seja, uma versão antiga. Com isso surge um problema: o usuário depende da velocidade com que a EAC integra a função cluster em novas versões do servidor do MySQL. A versão modificada do servidor monitora a porta padrão 3306 do MySQL e recebe por ali todas as solicitações.

O pacote de instalação do EAC é fornecido apenas no formato RPM, mas é possível instalá-lo, por exemplo, em sistemas Debian. Para isso é necessá-

rio descompactar os pacotes e executar manualmente os passos da instalação, o que significa um certo trabalho a mais. Mas não espere suporte técnico para a façanha – a não ser, é claro, que você pague à parte.

Além disso, em comparação com o servidor MySQL padrão, o EAC tem algumas limitações. Entre elas:

- ➔ suporta apenas `LOAD DATA LOCAL INFILE`. O comando `LOAD DATA INFILE` não é suportado.
- ➔ apenas os formatos de tabela InnoDB e o MyISAM são suportados.
- ➔ As funções `NOW()`, `CONNECTION_ID()` e `RAND()` só funcionam em acessos apenas para leitura.
- ➔ Os comandos `BEGIN`, `COMMIT` e `ROLLBACK` não podem ser utilizados.
- ➔ Os comandos `ANALYZE`, `KILL` e `OPTIMIZE` não são suportados.

Duas redes

O EAC usa uma rede aberta para o tráfego do MySQL e uma rede fechada para a troca de dados entre os membros do cluster e na administração. A interface aberta é totalmente administrada pelo software do cluster. Todos os nós utilizam os mesmos endereços IP e MAC para suas interfaces abertas de rede; nesses endereços o cluster pode ser acessado como se fosse uma entidade única. A base da estrutura de cálculo usada é de simples compreensão.

No momento podem ser unidos até 16 servidores em um cluster EAC e novos nós podem ser integrados ao sistema durante a operação. Os dados são sincronizados automaticamente. Para essa sincronização os modos *Blocking* e *Nonblocking* estão à disposição. No modo *Blocking* todas as tabelas são completamente bloqueadas durante a transmissão dos dados. No modo *Nonblocking* só são bloqueadas as tabelas que são transferidas. Uma limitação (não documentada claramente) é que

a versão *Nonblocking* funciona apenas quando o cluster contém pelo menos três servidores.

Tempo indesejado de inatividade

Se as tabelas contiverem grandes quantidades de dados o cluster não reage por um certo tempo durante o processo de sincronização. No teste foram copiadas tabelas bastante acessadas com mais de 20 milhões de dados, sendo que os programas que acessam esses dados não conseguiam iniciar consultas por vários minutos. Se a sincronização for realizada no modo *Nonblocking*, todas as consultas que incluem mais que uma tabela são evitadas. Isso garante a consistência dos dados.

Operação gráfica

Ao contrário do Cluster da MySQL AB, o EAC oferece uma interface gráfica para a supervisão e a administração (figura 4). A ferramenta está programada em Java e funciona de forma rápida e estável. Após

o Login no EAC seus nós aparecem em forma de anel. O símbolo do nó informa o seu estado.

Cada nó pode ser administrado em separado; os comandos correspondentes são transmitidos com um clique do mouse. Isso é válido apenas para os comandos do cluster – o servidor MySQL local no nó tem que ser administrado de outra maneira. A ferramenta também oferece a opção de mostrar estatísticas na forma de gráficos contínuos e de ver os arquivos de registro (*log*). São mostrados, entre outros, a carga, os tempos de resposta, as taxas de solicitações e a média da transmissão de dados. Todas as funções da interface gráfica – menos os gráficos com as estatísticas – podem ser acessadas através de uma ferramenta de linha de comando.

Conclusão

Não é fácil afirmar com certeza qual dos sistemas é o melhor, mas uma coisa é certa: o fato de o mercado estar crescendo é formidável. Já se pode prever que na área



Figura 4: Ao contrário da solução de cluster produzida pela MySQL AB, o produto da EAC oferece ao usuário uma excelente interface gráfica para a administração do sistema.

da alta disponibilidade e da tecnologia de cluster as soluções completas para o servidor MySQL poderão fazer concorrência aos grandes sistemas de bancos de dados comerciais.

Com o EAC for MySQL a Emic oferece uma solução de fácil manuseio, na qual os bancos de dados já existentes em servidores MySQL antigos podem ser transferidos sem problemas para o cluster. Efeito colateral (benéfico): a migração das aplicações também é sópa!

Além disso, o EAC está fazendo sucesso com uma instalação fácil em distribuições suportadas (entre elas várias versões SUSE Linux e Red Hat Enterprise). Apenas na configuração dos componentes ativos da rede que ligam à rede aberta são necessárias correções manuais ocasionais. Quando o EAC já estiver operando o sistema quase não necessita mais de ajustes do administrador.

Mas o administrador tem que fazer a manutenção do MySQL em todos os nós que compõem o cluster separadamente. Outra dificuldade é a sincronização de arquivos, principalmente os com grande quantidade de dados e que são consultados com frequência. A transferência

de dados pode causar longos tempos de parada, justamente o efeito que o uso do cluster quer evitar.

Com a interface gráfica é fácil verificar a carga do cluster. Mesmo usando máquinas com diferentes níveis de desempenho a carga fica bem distribuída.

Durante os testes ocorreram algumas quedas do sistema; entretanto a maioria delas não pôde ser reproduzida e sua causa exata não pôde ser apurada. Solicitações a respeito dos freqüentes problemas foram rapidamente atendidas e solucionadas com a ajuda da equipe de suporte técnico.

O Cluster da MySQL AB encontra-se ainda em uma fase ativa de desenvolvimento. Uma das áreas que causa dúvidas é a manutenção dos dados na memória RAM. Os desenvolvedores estão trabalhando ativamente em uma solução, que ainda não despontou, infelizmente, no horizonte.

A documentação está bastante ultrapassada, por isso recomendamos fortemente uma visita à lista de discussão do MySQL Cluster [2]. De qualquer forma, a configuração inicial e a inicialização do Cluster está muito bem descrita no manual do MySQL. A separação de ser-

vidores para a manutenção dos dados é uma boa solução; porém, os nós API serão o ponto fraco do sistema nesse caso, já que não podem executar *failovers* automáticos. Também não é possível simplesmente operar vários nós API no mesmo endereço IP.

A configuração do Cluster através do nó de administração facilita toda a administração da rede, porque os ajustes de memória dos diferentes nós de bancos de dados podem ser feitos de forma centralizada. Entretanto formato NDB traz algumas limitações a esse processo. ■

INFORMAÇÕES

[1] Emic Networks www.emicnetworks.com

[2] Lista de discussões do MySQL Cluster: lists.mysql.com/cluster

[3] MySQL AB: www.mysql.com

SOBRE O AUTOR

Michael Spindler é desenvolvedor de sistemas do canal privado de meteorologia MC-Wetter em Berlim. Além de atuar no desenvolvimento do sistema Linux, ele trabalha principalmente no serviço de alerta de tempestades.

Tipos de tabelas no MySQL

O MySQL é hoje composto por quatro *storage engines* principais: *MyISAM*, *InnoDB*, *Cluster* e *MaxDB*. Um projeto pode utilizar desde um banco de dados simples, como o *MyISAM*, até um banco de dados corporativo, como o *MaxDB* (antigo *SAP DB*), que suporta qualquer aplicativo SAP já há muitos anos com suporte a *triggers*, *views* e *stored procedures*. Todos são licenciados sob a GPL.

Existem várias propriedades do Sistema Gerenciador de Banco de Dados (*SGDB*) que independem do *storage engine* utilizado, tais como a sintaxe de comandos, mecanismos de acesso, *logs*, dentre outras. Por outro lado, recursos como capacidade transacional (*commit* e *rollback*), níveis de isolamento e integridade referencial, estão presentes apenas em alguns *storage engines*. Assim, pode-se optar por um ou outro mecanismo dependendo dos requisitos da sua aplicação ou até de cada tabela da mesma base de dados.

Poderíamos classificar esses *storage engines* seguindo dois critérios: o primeiro associado ao tipo de armazenamento dos dados (memória ou disco), e o segundo associado à capacidade transacional (veja o quadro ao lado).

É possível implementar um sistema que faça uso de todos os *storage engines* apresentados. Em relação ao *MySQL Cluster*, apenas as tabelas criadas com o tipo *Cluster* serão inseridas no agrupamento. Tabelas pré-existentes com tipos como *MyISAM* e *InnoDB* são ignoradas. Já o *EAC Cluster*, que não é livre, suporta esses formatos, mas com a limitação de utilizar uma versão especial do binário do servidor MySQL. Existem ainda outras soluções de cluster para MySQL, como o da Veritas, que utiliza armazenamento externo.

Antes de pensar na criação de clusters MySQL, sugerimos explorar ao máximo os recursos de replicação, que são de fácil implementação e extremamente confiáveis. A correta escolha dos tipos das tabelas, aliada a um ajuste fino correto do sistema promove grandes melhorias no desempenho.

Por Eber Duarte – DBA certificado MySQL Core e Professional e Helvécio Borges Filho (www.mysqlbrasil.com.br)

Storage Engines do MySQL

	Em memória	Em disco
Transacional	Cluster	InnoDB, MaxDB
Não transacional	MEMORY	MyISAM