

Versões futuras do Firebird....

Existiram várias discussões a respeito do futuro do Firebird dentro do grupo de administradores. Com base nestes comentários e no retorno obtido, acreditamos que chegou a hora de abrir a discussão para uma audiência mais ampla, para seu retorno e comentários.

O roteiro de desenvolvimento proposto segue abaixo. Note que não é definitivo, e pode ser modificado e adicionado com base na sua opinião.

Atualmente temos 3 ramos de código do Firebird, a versão corrente do código 1.5, Firebird HEAD, a base para a próxima versão e o Vulcan.

Existe espaço para a liberação da versão 1.5.2 se alguma regressão for encontrada na versão 1.5.1

Desejamos parar novos desenvolvimentos na HEAD, e usar o código para criar a versão 1.6 do Firebird. Então fazer a união do Firebird 1.6 com o Vulcan para criar o Firebird 2.0.

Para os que não conhecem todos os detalhes, HEAD é o ramo atual de desenvolvimento dentro do CVS na sourceforge.

Vulcan é um ramo experimental que o Jim Starkey tomou como base da HEAD há 6 meses para implementar multi-threading com alta granularidade (suporte a SMP) e suporte aos sistemas operacionais de 64bits tendo como base um contrato de desenvolvimento que a IBPhoenix possui com um grande fornecedor de software.

Tanto o Firebird 1.6 quanto o Vulcan estarão provavelmente disponíveis para testes na mesma época, i.e. provavelmente existirão "builds" que os usuários podem instalar e usar em simultâneo. Essencialmente durante este tempo os usuários que utilizarão as versões antes do merge terão que escolher entre a utilização de SMP (Vulcan) e as novas funcionalidades do FB (FB 1.6).

A razão por traz disso é que existe um sentimento de que se tentarmos fazer tudo ao mesmo tempo (2.0 = HEAD + Vulcan), é que isto levará muito tempo antes que o projeto possa liberar a próxima versão do Firebird, o que não é considerado uma coisa boa para o projeto.

O roteiro é o seguinte:

Todas as funcionalidades incompletas devem ser finalizadas.

Funcionalidades da versão 2.0 devem ser implementadas o quanto antes para a inclusão das funcionalidades do YAFFIL e o trabalho do Paul Ruizendaal.

HEAD - Versão alpha inicial acontecerá aproximadamente em Setembro.

Versão final no fim do ano ?

VULCAN - Versão alpha inicial deve acontecer num futuro próximo. Futuras versões betas assim que disponíveis.

Versão final no fim do ano ?

FIREBIRD 2.0 - Junção da versão 1.6 com o Vulcan, versão alpha Abril/Maio de 2005, versão final no fim de 2005.

Todas as datas são estimadas, não existe garantia que qualquer data mencionada acima é alcançável. Mas podemos tentar.

A versão 1.6 irá requerer uma versão do ODS compatível com as versões futuras...

Então, o que existe nas versões HEAD, YAFFIL e Vulcan ? E qual o trabalho feito por Paul Ruizendaal ?

HEAD

Derived Tables (Tabelas Derivadas)

Uma tabela derivada é aquela que é criada on-the-fly, usando uma sentença SELECT, e referenciada como uma tabela normal ou uma view.

Tabelas Derivadas existem somente em memória e podem ser referenciadas por um SELECT externo (outer SELECT) na qual foi criada. Elas não são tabelas temporárias. Um exemplo poderia ser:

ex. Criar uma listagem de gerentes (managers) cujos departamentos tenham mais que 25 funcionários (employees)...

```
Select
    e.name, d.department
from
    employees e
    join (select
            dl.department, dl.mgr_id
        from
            departments dl
            join employees el on (dl.dept_id = el.dept_id)
        group by
            dl.department, el.emp_id
        having
            count (*) > 25
    ) d on (d.mgr_id = e.emp_id)
```

Versão Proposta: 1.6

Nova estrutura de índice

Opinião:

O trabalho feito consiste em um novo layout de página B-tree, isto tipicamente implica em uma mudança significativa de ODS. Consideramos este trabalho crítico para a próxima versão do Firebird. Então precisamos encontrar uma solução para isso.

Versão Proposta: 1.6

Auto union-cast

Versão Proposta: 1.6

Seletividade do índice baseada nos segmentos (índices compostos)

Arno Brinkman e Dmitry Yemanov fizeram todas as mudanças necessárias na ODS. O Otimizador precisa ser melhorado para acomodar as novas estatísticas. Incompleto.

Versão Proposta: 1.6

Sintaxe EXECUTE BLOCK

Versão Proposta: 1.6

Reimplementação da sintaxe do cursor em PSQL

Versão Proposta: 1.6

Sintaxe CANCEL STATEMENT

Incompleto

Versão Proposta: 1.6

Implementar backup incremental físico e multi-nível.

Nbackup precisa ser confirmado como completo, documentado e testado.

Versão Proposta: 1.6

Modo de manutenção de banco de dados monousuário e desligamento completo (Single-user database maintenance mode and full database shutdown)

Dois novos modos de desligamento do banco de dados:

multi-user maintenance mode (modo de manutenção multi-usuário) - somente o SYSDBA pode acessar o banco de dados, atividade do Garbage Collector é inibida (desligamento como implementado atualmente no Firebird 1.0.x).

single-user maintenance mode (modo de manutenção mono-usuário) - somente uma conexão do SYSDBA é permitida ao banco de dados.

complete database shutdown (desligamento completo) - todas as conexões são negadas a não ser que seja pra colocar o banco de dados novamente on-line.

Nomes sugeridos para as chaves:

gfix -shut -complete ...

gfix -shut -single ...

-single/-complete podem ser usados em combinação com todas as modos de desligamento atuais.

Versão Proposta: 1.6

Condições de pouco espaço em disco

O gerenciador de arquivos temporários funciona adequadamente, mas o gerenciador de páginas é um ponto de falha. Se não existir espaço suficiente para armazenar uma nova página, então o banco de dados pode se corromper. Se o "método de escrita cuidadosa" (careful write) estivesse funcionando adequadamente isto não deveria ser problema...

Incompleto

Versão Proposta: 1.6

Windows local connect

Torcar o protocolo local de IPC para XNET

Versão Proposta: 1.6

YAFFIL

Incorporar as melhorias feitas no otimizador do Yaffil

1. Execução otimizada de queries IN (<lista>) (e múltiplas igualdades usando OR, claro, pois é a mesma coisa)

2. Permitir que sentenças com WHERE A = <value> ORDER BY B usem o índice composto (A, B)

3. Outras correções e bugfixes do otimizador

Versão Proposta: 1.6

Expression Indices (Índices baseados em expressões)

Versão Proposta: 1.6

Redução de escala para numéricos e decimais (numerics e decimals)

Quando a precisão do resultado é maior que MAX_PRECISION, a escala será reduzida para evitar que a parte inteira seja truncada.

Atualmente: $D(18, 1) * D(18, 17) = D(18, 17)$ // Parte inteira com apenas um dígito !?

Deveria ser: $D(18, 1) * D(18, 17) = D(18, \text{FIXED_PRECISION})$ // FIXED_PRECISION ou será hardcoded ou configurável.

MSSQL e o Sybase definem como 6 (mas a precisão máxima deles é 38).

O Yaffil implementa esta funcionalidade de duas formas: DPB flag e o tipo de dado não padrão __MONEY. Nossa própria implementação continua a ser o assunto de uma discussão à parte.

Versão Proposta: 1.6

Implementar uma lista de de 40 funções internas.

Funções incluem tudo da `ib_udf`, `fbudf`, incluindo as funções "SQL compliant" `TRIM/PAD/CHAR_LENGTH` etc., outras rotinas de conversão e até mesmo expressões regulares.

Opinião:

Será disponibilizados através de BLR ? Se for, usarão novos valores BLR ?

Versão Proposta: 2.0

Configuração de time-out por sentença

Versão Proposta: 1.6

VULCAN

Suporte a plataformas de 64-bits

Versões para Linux de 64 bits, e outros sistemas operacionais.

Suporte a SMP e Hyper Threading no Windows

Detalhes:

Deixar a versão SuperServer totalmente multi-threaded e SMP-friendly.

Conexão Windows local

IPC para XNET

Implementar suporte total para rodar mecanismos concorrentes (concurrently running engines) - Win32

Extender Instalador Windows.

Implementar um serviço de console MMC para permitir ao projeto e a terceiros a criação ferramentas de gerenciamento para o Firebird e um lugar para integra-los.

O mesmo para Linux ?

É possível fazer o equivalente em Linux ? Sim

Processo de desligamento limpo para servidores Classic e processos em Win32.

Implementar mutexes para que possamos fechar de modo programático e fácil servidores Classic. Nós podemos então integrar controle para Classic Server no applet do Control Panel.

Desligamento limpo para processos Classic em ambientes POSIX.

Atualmente, a única opção é dar um "kill" no processo.

Arquivos de configuração por banco de dados

Gateways para InterBase e Servidores Firebird legados.

Uso de SQL Interno

Cache de sentenças compiladas (o alicerce está pronto)

Tabelas de Sistemas em camadas (Layerable system tables)

Mecanismo Unificado (unified engine) (superserver/classic/embedded)

Arquitetura extensível (providers)


OUTROS

Ferramenta GUI simples e multi-plataforma

Versão Proposta: 1.6

Saudações !

Projeto Firebird

<p>Artigo Original:</p> <p>Projeto Firebird (Paul Beach)</p>	
<p>Tradução e adaptação:</p> <p><u>Alexandre Benson Smith</u> alexandre.smith@comunidade-firebird.org</p>	<p>Comunidade Firebird de Língua Portuguesa</p> <p>Visite a Comunidade em: http://www.comunidade-firebird.org</p>

A Comunidade Firebird de Língua Portuguesa foi autorizada pelo Autor do Original para elaborar esta tradução.