

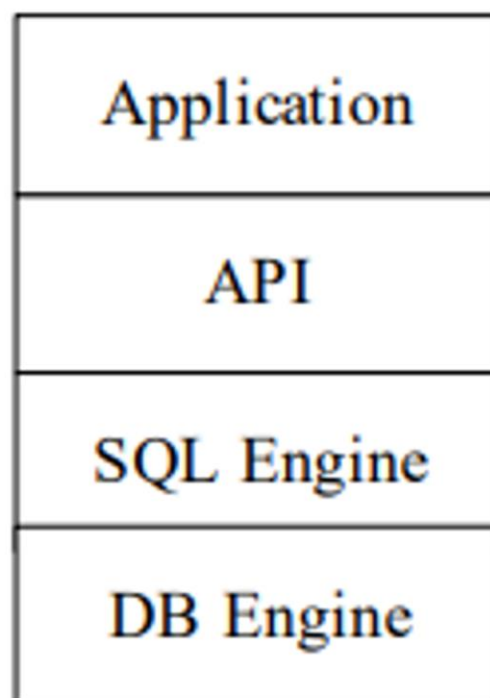
## 3. System Architecture

Este capítulo apresenta em detalhes os dois modelos de arquitetura do DBMaker. Examinaremos primeiro o processo DBMaker e a Área de Comunicação e Controle de Banco de Dados (DCCA), que armazenam todas as informações necessárias para cada banco de dados iniciado. Em seguida, a arquitetura de ambos os modelos será explicada.

### 3.1 The DBMaker Process

Um processo do DBMaker gerencia o armazenamento e a recuperação de dados de acordo com os comandos do usuário e outras funções do banco de dados. Um processo do DBMaker consiste em várias camadas, conforme mostrado na Figura.

A Figura ilustra as aplicações de usuário se comunicando com o DBMaker através de uma Interface de Programação de Aplicações (API). A API transmite os comandos do usuário (ou seja, comandos SQL) ou chamadas de função para o Motor SQL. O Motor SQL é responsável por analisar e traduzir os comandos SQL em sequências de chamadas de função que são aceitáveis para o Motor de Banco de Dados. Em seguida, o Motor SQL passa essas chamadas para o Motor de Banco de Dados, que executa essas chamadas de função para armazenar dados em tabelas ou recuperar dados das tabelas.



O Mecanismo de SQL e o Mecanismo de Banco de Dados possuem funções diferentes. O Mecanismo SQL lida com a análise de SQL e a otimização de consultas. O Mecanismo de Banco de Dados gerencia espaço/buffer, controle de concorrência, recuperação de falhas e outras tarefas relacionadas. Todos os módulos colaboram para manter a consistência dos dados em todo o banco de dados. A maioria dos parâmetros de ajuste de desempenho está relacionada ao Motor de Banco de Dados.

A API e os Mecanismos SQL são idênticos nos modos de usuário único e cliente/servidor. No entanto, os Mecanismos de Banco de Dados nos modos de usuário único e cliente/servidor são diferentes. O modo de usuário único pode atender apenas a um usuário, enquanto o modo cliente/servidor pode atender a múltiplos usuários.

No modo cliente/servidor, a aplicação e a API estão interligadas e são executadas nas máquinas cliente, enquanto o Mecanismo SQL e o Motor de Banco de Dados estão interligados e são executados nas máquinas servidor. Dessa forma, a API pode se comunicar com o Motor SQL por meio de protocolos de rede.

## 3.2 Database Communication and Control

### Area (DCCA)

Quando iniciado, o DBMaker aloca primeiramente um grande bloco de memória para armazenar informações relacionadas ao banco de dados, como pools de buffer e vários tipos de informações de controle. Esse bloco de memória é chamado de Área de Comunicação e Controle do Banco de Dados (DCCA). Ele contém três tipos de dados: ***page buffers, journal buffers, and the System Control Area (SCA)***

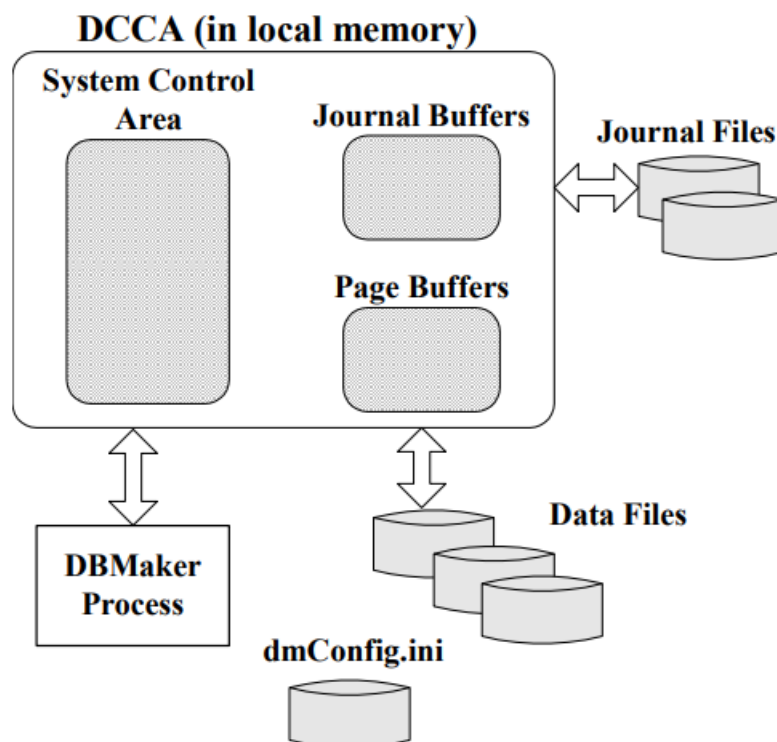
A DCCA é muito importante para o funcionamento do DBMaker, especialmente quando executado no modo client/server. A DCCA é alocada a partir da heap privada para ambientes de usuário único do Microsoft Windows e UNIX. Em um ambiente client/server UNIX, a DCCA deve ser compartilhada entre todos os processos do DBMaker que acessam o mesmo banco de dados, portanto, não pode ser alocada a partir da heap privada. Em vez disso, é utilizado o mecanismo de memória compartilhada do UNIX para alocar a DCCA. Todos os processos do DBMaker que são executados no modo client/server comunicam-se entre si via DCCA.

## 3.3 Architecture of the Single-User Model

O modo de usuário único do DBMaker é um sistema de gerenciamento de banco de dados (SGBD) que suporta apenas um usuário ou aplicativo. É menor e mais rápido

do que outros modos, em parte, porque o controle de simultaneidade é desnecessário. O modo de usuário único do DBMaker é uma boa opção quando um usuário ou aplicativo é o proprietário do banco de dados. A Figura 3-2 ilustra a arquitetura do sistema do modo de usuário único do DBMaker.

Como apenas um usuário ou aplicativo pode se conectar simultaneamente a um banco de dados DBMaker de usuário único, a DCCA é obtida do heap privado e não é compartilhável. É importante observar que o DBMaker não oferece suporte a um mecanismo de bloqueio no modo de usuário único. O motor DBMaker aumenta o desempenho mantendo todos os dados do banco de dados na memória durante a execução e grava as páginas modificadas de volta em arquivos de disco, incluindo arquivos de dados e arquivos de log, nos momentos apropriados. O arquivo de texto dmconfig.ini define muitos parâmetros necessários para a configuração do DBMaker.



### 3.4 Architecture of the Client/Server Model

O DBMaker também oferece suporte ao modo client/server. Nesse modo, o programa de aplicação é composto por dois processos: o processo do aplicativo client e o processo do server de banco de dados (também chamado de processo do servidor). Normalmente, o processo cliente reside em um PC ou estação de trabalho front-end e usa rotinas de biblioteca de API do DBMaker para se comunicar com o processo do servidor. O processo do servidor está localizado em outro lugar como parte de uma

rede local. É importante observar que, em uma configuração client/server, todos os computadores envolvidos, incluindo server e client, podem ser de tipos de plataforma diferentes.

A versão client/server do DBMaker inclui um módulo de gerenciamento de rede. Este módulo deve ser instalado tanto no client quanto no server. Os gerenciadores de rede são responsáveis por enviar dados entre os clientes e os servidores de banco de dados. O protocolo de comunicação de rede é importante no modelo client/server. O DBMaker atualmente suporta apenas TCP/IP (Transmission Control Protocol/Internet Protocol). Ao executar a versão client/server do DBMaker em um sistema que normalmente não suporta TCP/IP, é necessário instalar o software de rede TCP/IP antes de executar o DBMaker. Se o aplicativo client for executado no UNIX ou Windows, software TCP/IP adicional não é necessário, pois esses sistemas operacionais possuem suporte TCP/IP integrado. No Windows, basta especificar TCP/IP como um dos protocolos de rede e instalá-lo no sistema. A Figura 3-3 mostra a arquitetura do sistema DBMaker no modo client/server.

Em sistemas UNIX, quando um processo client se conecta a um server de banco de dados, o server de rede DBMaker bifurca outro processo do servidor para lidar com as consultas subsequentes. O processo original do server de rede continua aguardando conexões de outros client. O Windows NT é um sistema operacional multithread. A versão NT do server de rede DBMaker (dmserver.exe) é um programa multithread.

Portanto, quando um processo client se conecta a um processo server em execução em um sistema NT, o processo server DBMaker cria outra thread em seu espaço de processo para lidar com as consultas subsequentes. A DCCA é alocada na memória local, não na memória compartilhada. Sempre há apenas um processo do server DBMaker por banco de dados no Windows NT. À medida que mais sistemas operacionais adicionarem suporte a multithreading, o DBMaker incorporará multithreading em vez de bifurcação de processo quando possível. Pesquisas atuais indicam que programas multithread são mais eficientes do que programas multiprocesso.

No modo client/server, o DBMaker está associado a três componentes: o programa server, o programa client e a biblioteca client.

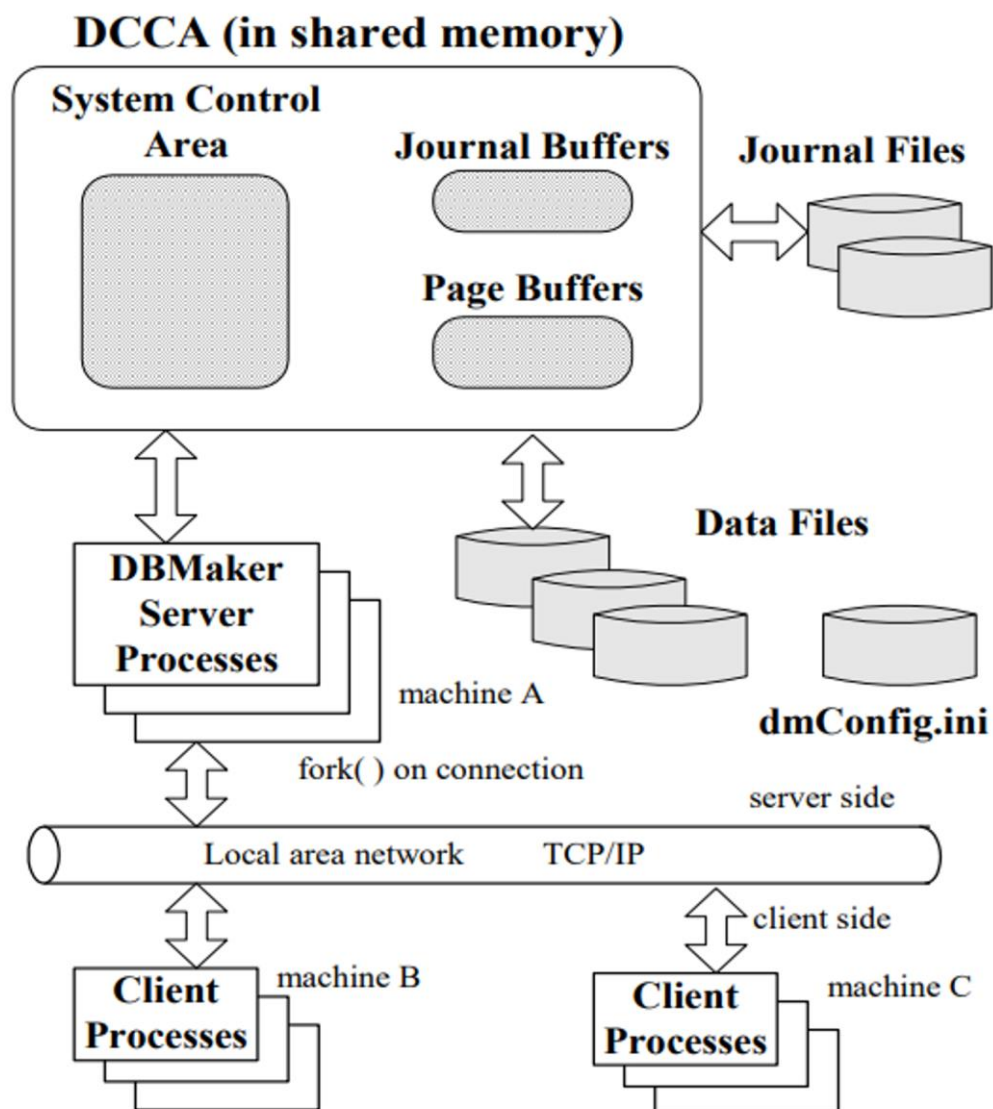
## **Server Program**

O programa server do DBMaker é chamado DmServer. Este programa inclui um gerenciador de rede que lida com a comunicação de rede e um mecanismo de banco

de dados que lida com o acesso a dados. Este programa deve ser iniciado primeiro para que os programas client possam se conectar ao servidor de banco de dados.

## Client Program

O programa client SQL do DBMaker é chamado dmsqlc. Este programa é usado para conectar um client a um banco de dados e, em seguida, emitir comandos SQL para processamento de dados.



## Client Library

A biblioteca client do DBMaker é chamada de libdmapic.a em UNIX, ou dmapi<numero\_da\_versao>.lib em sistemas Microsoft Windows. Os usuários que

planejam desenvolver seus próprios programas cliente devem vinculá-los com a biblioteca cliente. Por exemplo, os desenvolvedores podem usar várias ferramentas de desenvolvimento de diferentes fornecedores para escrever suas aplicações de interface. Ao construir as aplicações, eles devem vincular esses programas com a biblioteca cliente para que suas aplicações personalizadas possam se comunicar com o programa servidor.