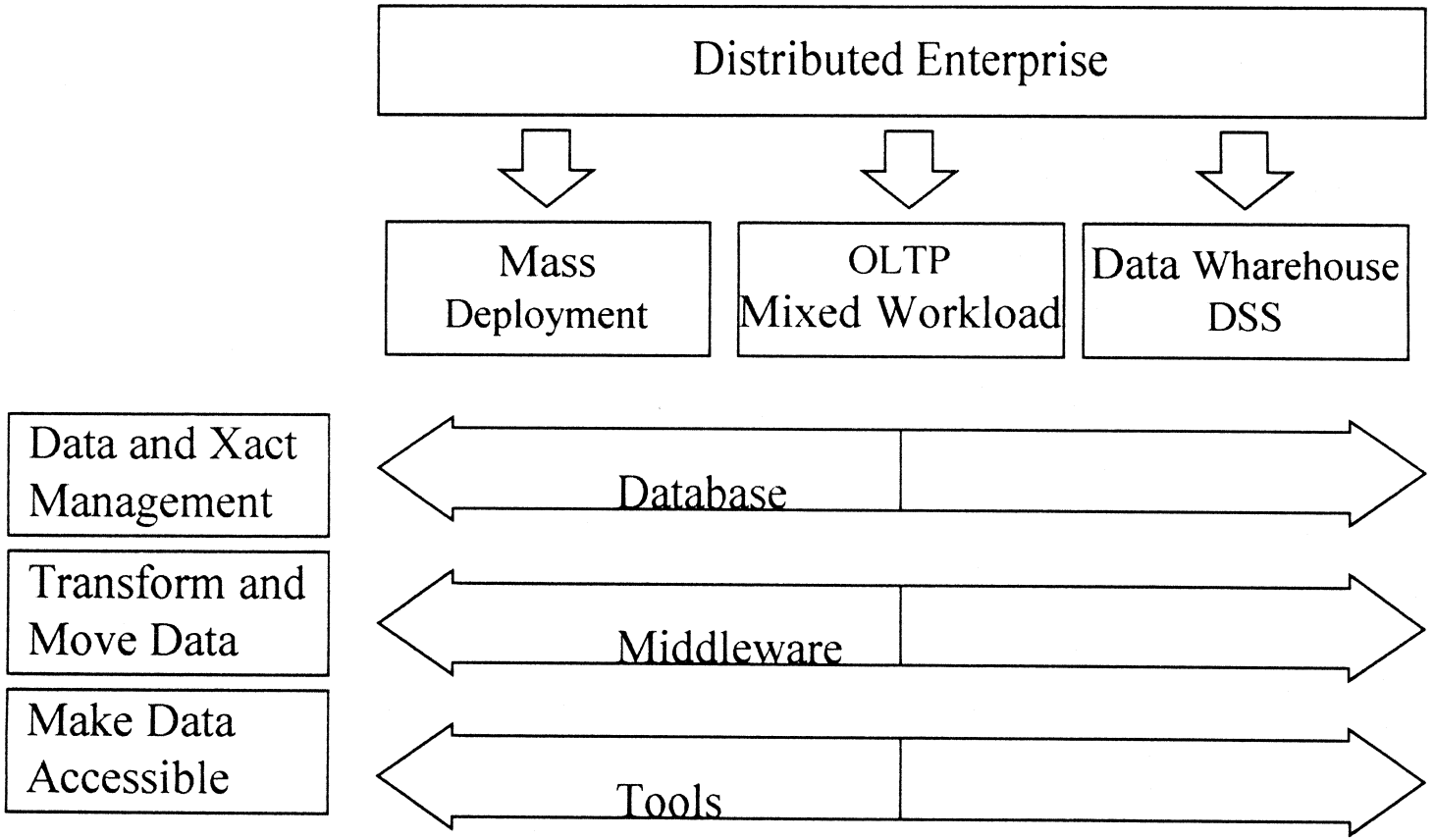
# SYBASE

# *Messages-Based computing*

Summary of the presentation "Message based computing" by Allen Lees at the "1995 International Sybase User Group European Conference

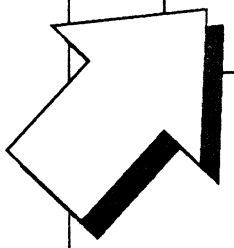
# Client/Server for the Distributed Enterprise



# The IT Challenge: extend into the Future

**Today**

- Desktop
- Wired networks
- Synchronous
  - wachten op resultaat
- Session-based



**Tomorrow**

- Desktop and mobile
- Wired and wireless
- Synchronous and asynchronous
- Session-based en messagebased

# What is messaging ?

- Messaging is the use of self-contained asynchronous computational elements which are passed from point of origination to point of receipt via zero, one, or more intermediate network points (nodes).

# What is a Message ?

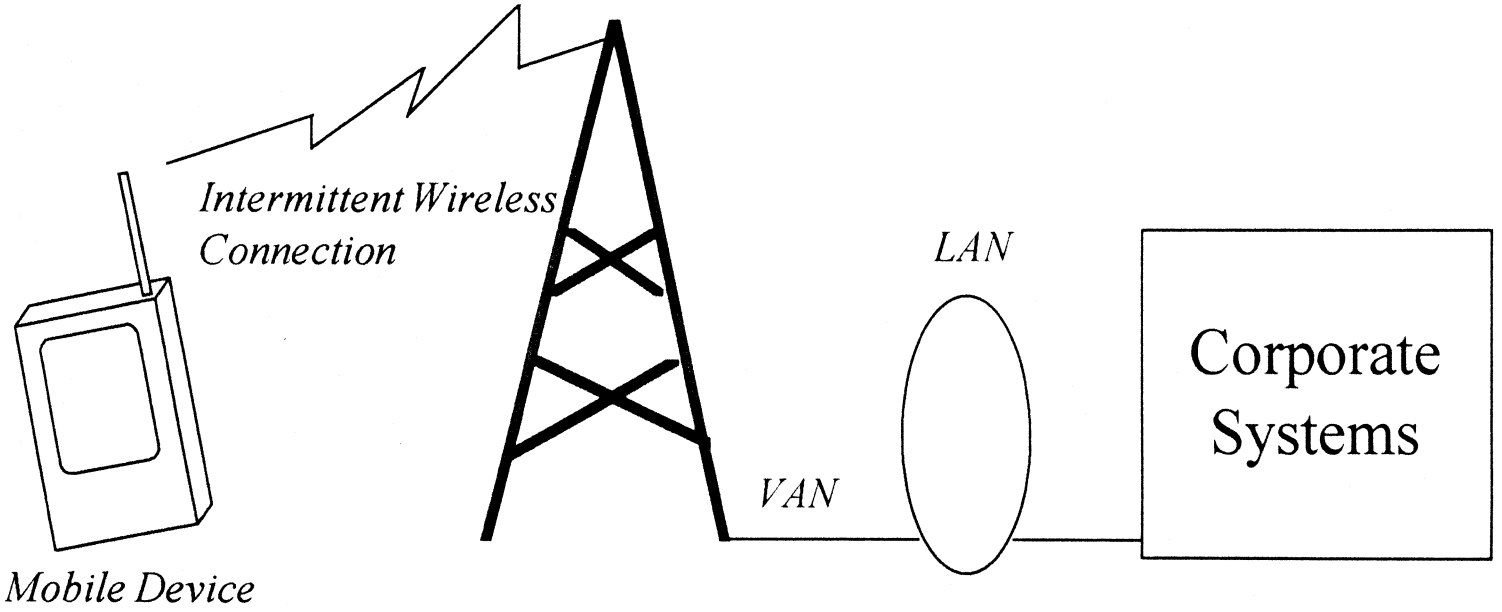
- Messages can be thought of as objects that move between clients and servers, containing instructions and data, without relying on maintenance of a persistent bi-directional connection between the client and the server. Messaging provides a connectionless service.

# Why Choose Messages ?

- Mobile computing
- Reduced-cost network-based computing
- Increased Efficiency of Resource Utilization
  - Interrupt-free / asynchronous computing
- Very large-scale client/server computing
  - Extended Enterprise Computing

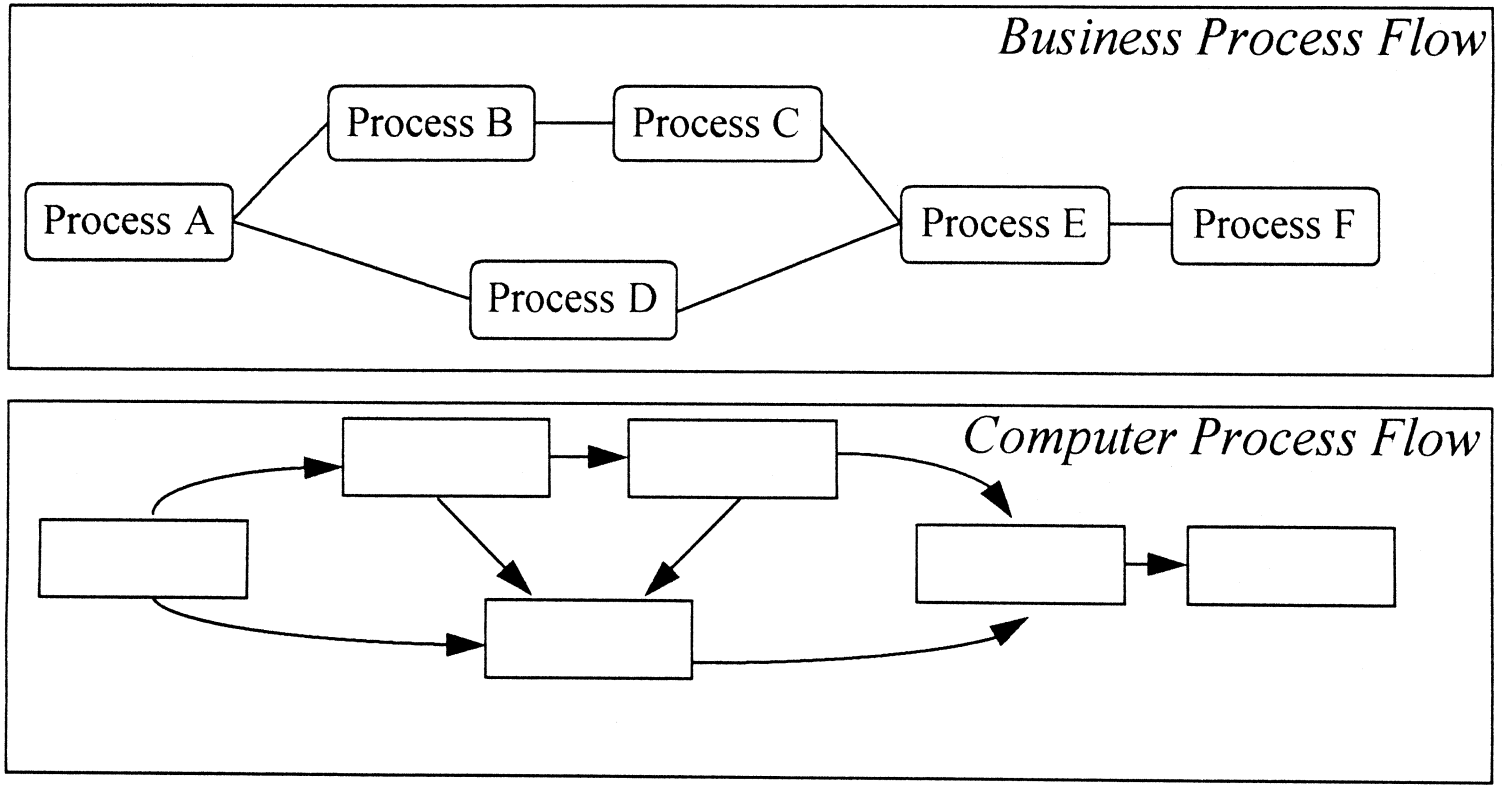
# Mobile Computing

- Reliable Delivery Over Unreliable Networks, Via Intermittent Connections



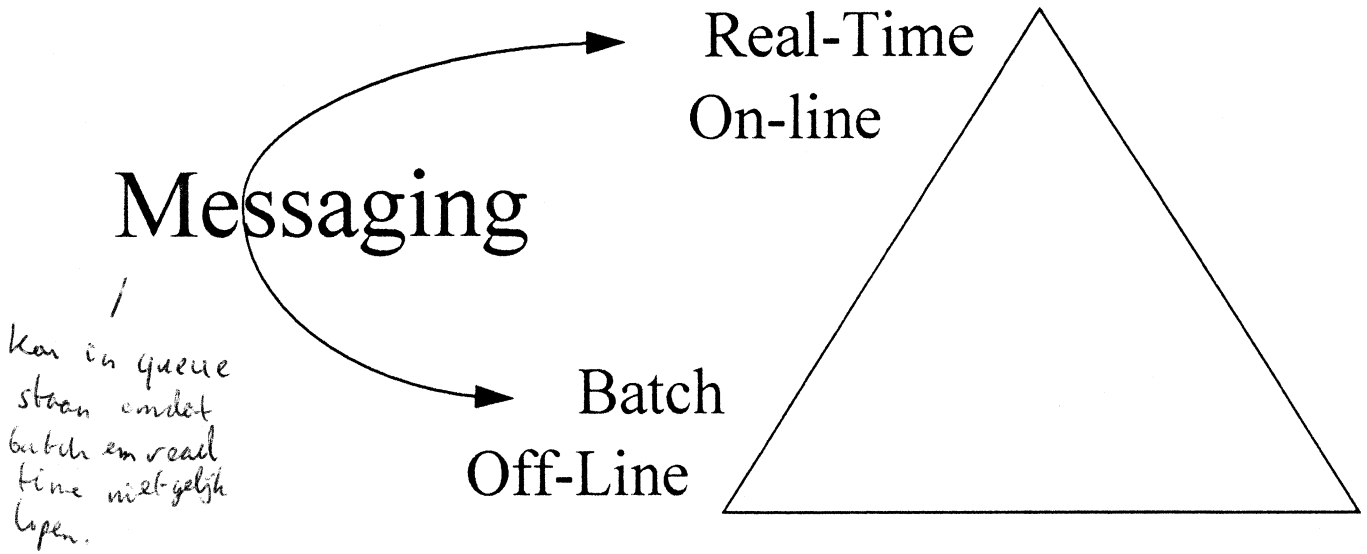
# Asynchronous Computing

- Matching Computing Process Flows To Business Process Flows



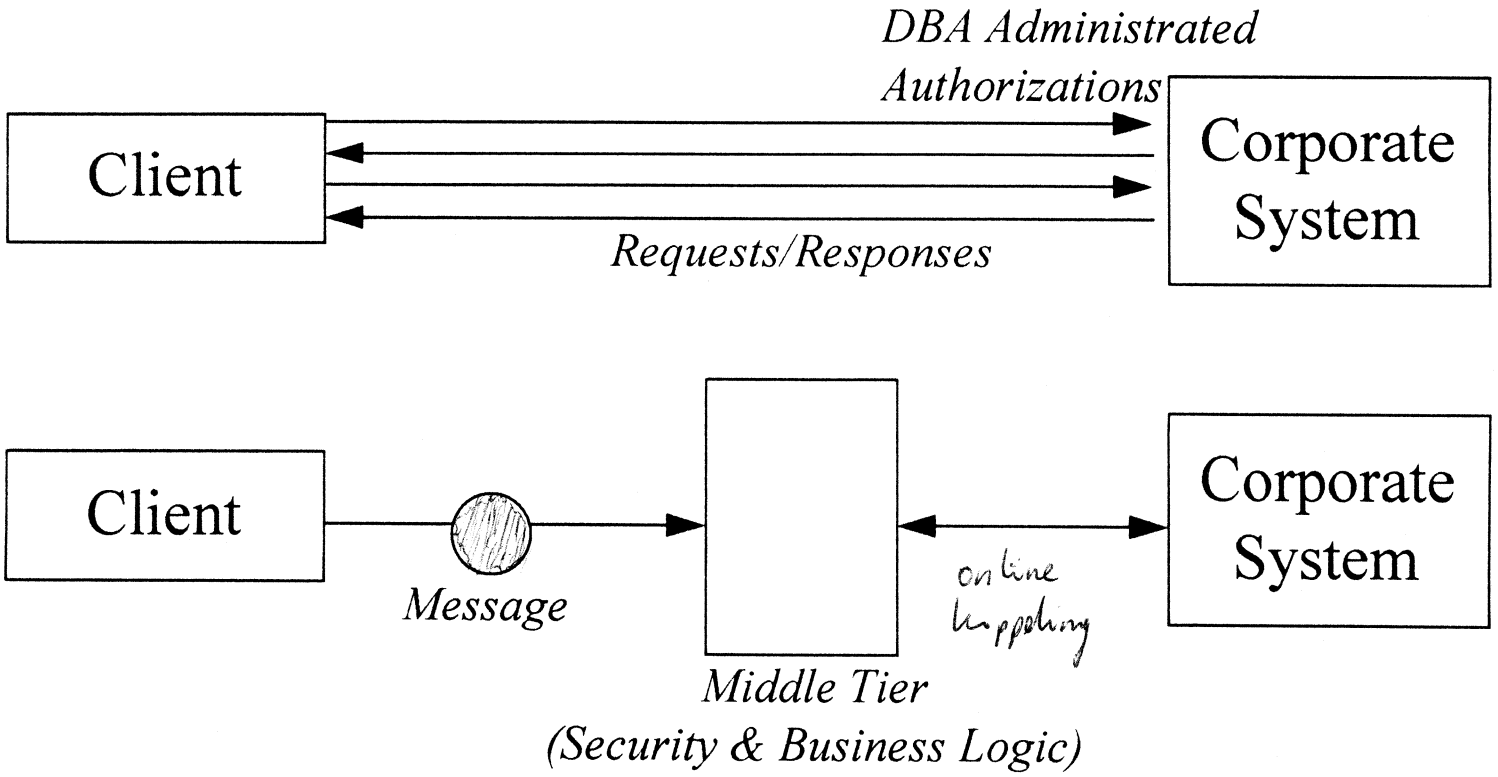
# Efficient Resource Utilization

- Moving Smoothly From On-Line to Batch



# Extended Enterprise

- Shifting Focus From Client Connections To Individual Messages

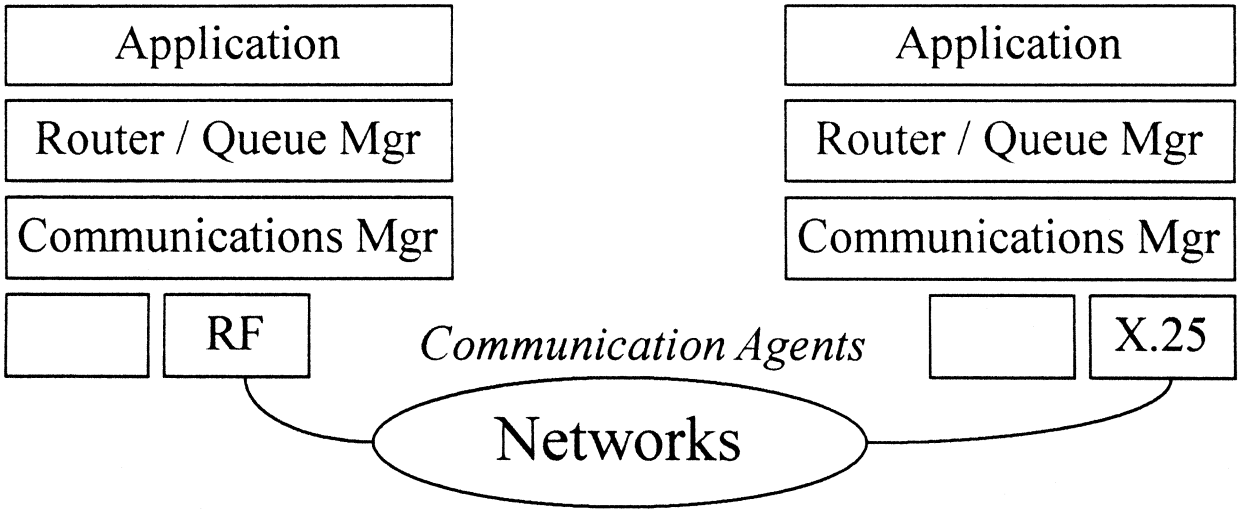


# Messaging infrastructure

- Routing Tables
- Store and Forward
- Authentication and Security
- Time to Live
- Broadcasting

# Sybase Messaging products

- Sybase EMS 3.2 (available now)
  - Message-Based Transport System
  - Reliable Queuing, Store&Forward, Routing, Recovery



# Mobile Devices

- Cost Effective Support For Mobile Devices
  - Small footprint
    - 120k - 190k
  - DOS and DOS PDA platforms
  - Smallest Windows and MAC's
  - Message protection from power loss
  - Unconstrained by message content

# Transparent Multiple Networks & Platforms

**Networks**

- Wireless
  - RAM
  - ARDIS
  - CDPD
  - Circuit-Switched Cellular
- Wired
  - TCP/IP
  - LU6.2
  - SPX
  - Appletalk
  - X.25

**Platforms**

- PC
  - DOS
  - Windows
  - Windows NT
  - MAC
  - OS/2
- UNIX
- Mainframe CICS/MVS

# Intelligent Quality of Services

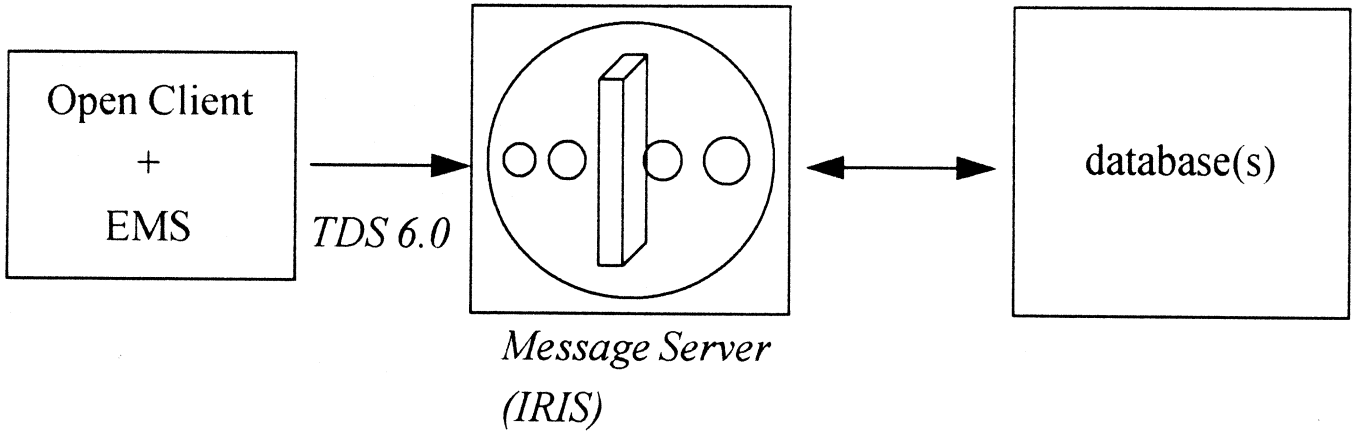
- Message Ack *"message received at 4:00 p.m."*
- Cost/Speed *"Route this cheaply"*
- Lifespan *"delete message after 10:00 p.m."*
- Priority/Queuing *"Deliver this ahead of any others"*
- Guaranteed Delivery *"This message must be delivered"*
- Situation - Sensitive *"Don't deliver BLOB's over circuit switched cellular"*

# Sybase's message structure

- contains the following features:
  - Unique message identifier
  - Logical destination name
  - Quality of service value
  - Time to live value
  - Security / authentication key
  - Method identifier
  - Data

# Sybase Messaging products

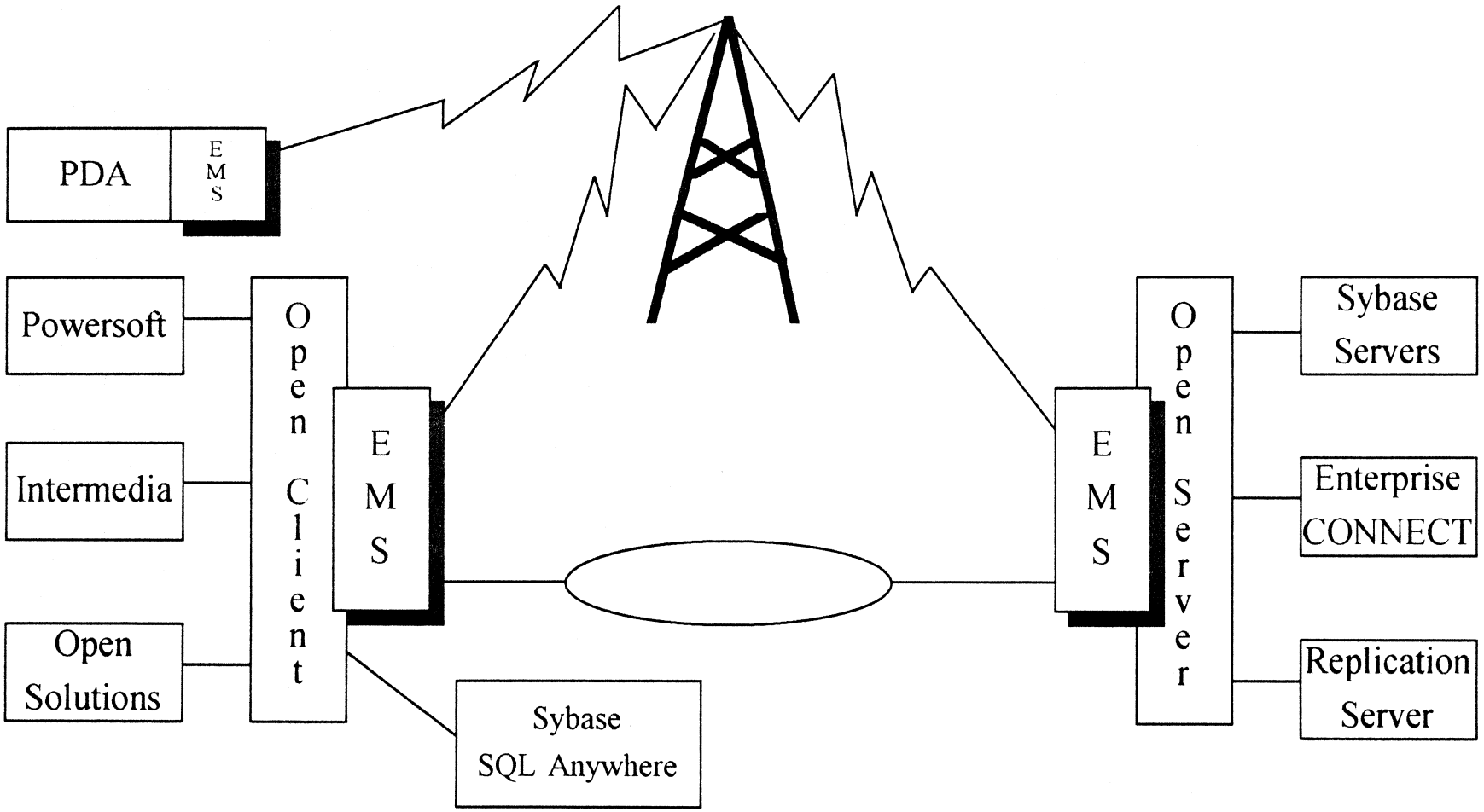
- Codename IRIS (available 1996)
  - Open Client / Server-Based Messing
  - Provides Data Marshaling, Lightweight TDS
  - Event driven, 3-Tier Architecture



# Messaging Strategy

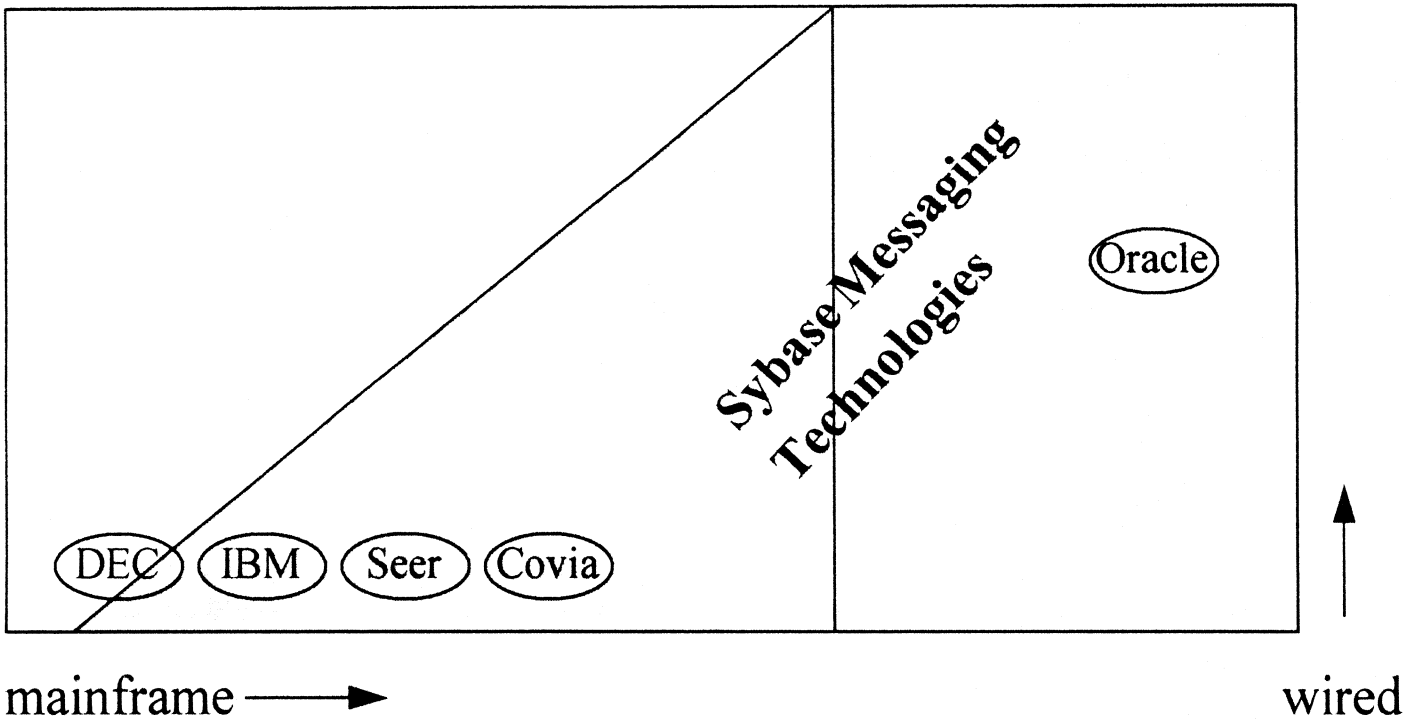
- Extension Of Client/Server From Sessions To Asynchronous Computing
- Part Of "Client Services" Approach
- Capabilities To Be Inherited Across The Architecture:
    - Replication Server
    - Enterprise Connect Gateways
    - Watcom SQL Database
    - Powerbuilder Tools

# Messaging Wherever Appropriate



# Sybase Competitive Positioning

- General Solutions, Not Point-Plays



# Availability

- Sybase EMS 3.2 Q3 95
- Sybase EMS 4,0 Q1 96
  - SMP Support
  - Additional PDA OSs
  - Additional public APIs
  - Embedded authentication
  - Additional networks
- OC/OS with Sybase EMS Q2 96

# Questions ?

# Sybase User Conference 1995

# REPLICATION SERVER

# What is data replication?

- The process of managing copies of data
- Replicate copies can be for any use and can be different from the source

- tight consistency
  - no latency - real time consistency is maintained
  - synchronous replication - requires two phase commit
- loose consistency
  - some latency exists before copies are consistent with a primary source
  - asynchronous replication - usually based on store and forward paradigm

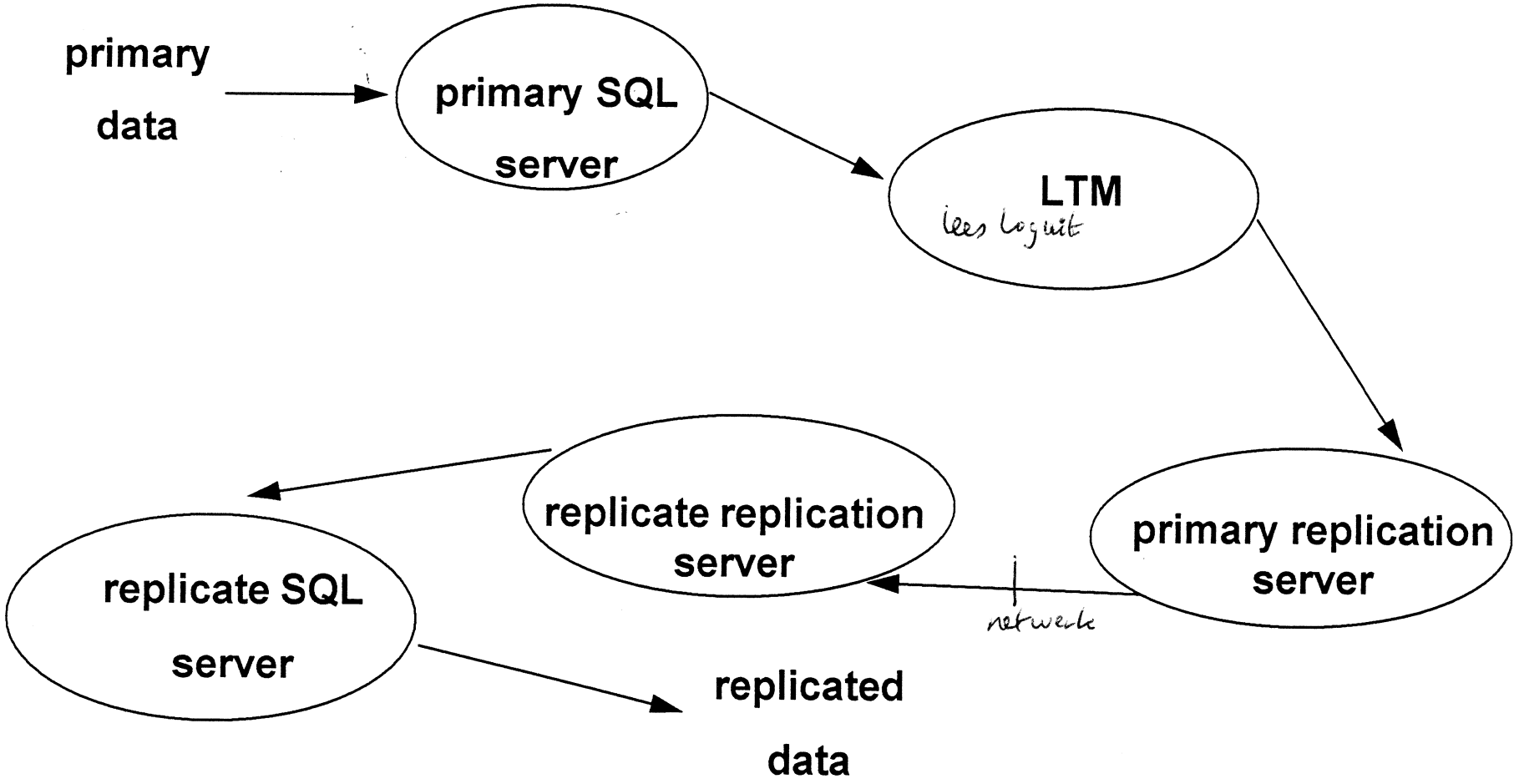
# Why replicate data?

- to improve data availability and performance in a distributed environment
  - eliminate single point of failure
  - bring data to local users
  - implement mobile computing
- to aid in disaster recovery - use as a warm standby
- to implement an OLAP environment separate from heterogeneous operational OLTP(s)

# Evolution of data replication

- Database dump and load
- Transaction log dump and load
- Table level dump and load (BCP)
- Snapshot-based replication
- Two-phase commit
- Sybase Replication Server

# Replication process



# Sybase Replication Server

- Market leader in data replication technology
- 550+ Installations
- 400+ Production Installations

**Proven, reliable technology for replicating data in near-real time throughout the enterprise.**

# Speakers

- **Marie Buretta**
  - president Marie Buretta Inc. → Chase Manhatten Bank
- **Perry M. Stufflebeam**
  - Spoornet Information Systems Zuid Afrika
- **Alan Buis**
  - National Center of Cinematography Frankrijk
- **Mikhail Boz**
  - Sybase
- **Ronald Olshausen**
  - Sybase

# Design Issues

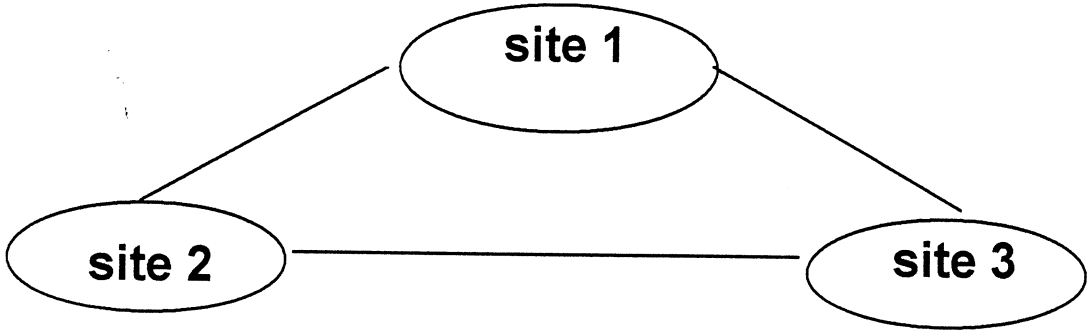
- Ensuring single primary sources
- Challenges with 'update anywhere model'
- Data server configuration issues
- Deciding between data and stored procedure replication
- Creating your own 'function string classes'
- Trigger Concerns
- Nested Transactions
- Updating primary keys
- Miscellaneous issues

# Ensuring single primary sources

- Every piece of data is owned and updated at a single site within a designated time frame
  - primary sources can be distributed or centralized
  - ownership of an element can be changed

Must be designed and implemented into your application systems by you

# Challenges with 'update anywhere model'




initially all sites have row a at state 0 and row b at state 0

route between site 1 and site 3 goes down

site 1 updates a to 1 and b to 1

the updates are replicated to site 2

site 2 updates a to 2

update is replicated to site 1 and site 3

then the update from site 1 is replicated to site 3

# Data server configuration issues

- Release and fix level (EBF) for primary and replicate sites
- Log attributes (size truncation parameters)
- Data types utilized in the replication proces
- Database mirroring
- Sort orders
- Character set usage

- Log groot genoeg maken

# Deciding between data and stored procedure replication

- When to use table replication?
  - where quantity of table changes are low
  - to maximize application transparency
  - where simple mapping compensates for denormalization at replicate site(s)

- When to use function replication?
  - to minimize WAN traffic
  - to execute different procedures at primary and replicate sites
  - where complex mapping must be done to compensate for schema differences

# Creating your own 'function string classes'

**default function string class is rs_sqlserver_function_class**

## Advantages of using this class

- **can take advantage of only sending data values that have changed**
- **get new release system functions by default**

## Disadvantages

- **no intrinsic 'row not found' checking (if @@rowcount != 0, no error is logged)**

# Trigger concerns

**Suppose we have a delete trigger on table a which inserts the deleted row(s) in table b**

| | | primary | replicate |
|---|---|---|---|
| before delete | table a | 1 | 1 |
| | table b | -- | -- |
| after delete | table a | -- | -- |
| | table b | 1 | 1<br>1 |

# nested transactions

replicatie van stored procedures

```
create proc P1 as
        update table 1
        begin tran
                insert table 2
                if @@error != 0 rollback tran
                else commit tran
        return
```

| non replicated s.p. | replicated s.p. |
|---|---|
| duplicate key in table 2 | duplicate key in table 2 |
| update table 1 committed<br>insert table 2 rollback | update table 1 rollback<br>insert table 2 rollback |

# Updating primary keys

## Client applications should not update primary key columns in multiple rows of a replicate in a way that could duplicate the key of another row

update table 1

set pkcol1 = pkcol1 + 1

| | |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |

LTM recieves the log records and submits the following commands to RS

set pkcol1 = 2 where pkcol1 = 1

set pkcol1 = 3 where pkcol1 = 2

set pkcol1 = 4 where pkcol1 = 3

| | |
|---|---|
| 1 | 4 |
| 2 | 4 |
| 3 | 4 |

# Miscellaneous issues

- Time implications
  - time information generated by "getdate()" on a SQL Server are local to the time zone of the machine
  - a solution: maintain table of offsets from GMT for each SQL Server
- Unique number generation
  - number will only be unique within each primary site not across sites
  - solutions
    - » composite key that incorporates the site identifier
    - » use a remote call to a unique key generator service
    - » assign large number ranges across primary sites
- Non logged statements
  - non-logged modifications to tables (truncate table or bcp) cannot be replicated

# Institutionizing a replication service in your firm

- Get appropriate training for individuals
- Establish firmwide standards
  - subscriber is not allowed to modify data at his site.
    subscriber is not supposed to act as a provider
- Sample meta data requirements for
  - table level information
  - column level information
- Sample naming standards for
  - replication server objects
  - replication server user accounts
  - replication server / data server objects
  - replication server / data server user accounts

# Institutionalizing a replication service in your firm (cont.)

- **Golden Rules**
  - Avoid making the implementation more complex than necessary
  - Ensure the implemention model resolves the business problem
  - Abide by all standards defined in your RS best practice guide
  - Analyze all site requirements carefully with respect to:
    - » number of users
    - » transaction volume
    - » availability of production support services

# Replication Server 11.0 features

- **Warm standby**
- **Trigger on/off option**

# Warm standby database objectives

- Replicate database - Consistent copy of the data
- Loose consistency is sufficient
- Easy to set-up and initialize
- Can easily replace primary database at crash time
- Minimal data reconsiliation is required
- Low impact on primary database

# Warm standby with RS 10.0

- Create replication definitions for each table being replicated
- Create subscriptions for individual tables
  - Load replicate tables using subscription materialization
  - Load replicate tables using bcp
- Serializing transactions
- Administrative requirements
  - Disabling triggers
  - Disabling LTM truncation

# Warm standby with RS 11.0

- **Easy initialization and setup**
  - Easy initialiation with an active system
  - No replication definitions required
  - No subscriptions required
- **Easy fail-over from primary to standby**
  - Automatic failover
  - Minimal manual reconciliation of before / after failover transactions

# Limitations in 11.0 Warm standby

- Applications must connect to the currently active primary
- Warm standby versus hot standby
- No meta-data replication geen tabellen aanmaken
- Primary and standby databases must be controlled by the same replication server

# Trigger on/off option

- Instead of using the maintenance user as in RS 10.0
- No automatic trigger generation when the replicate table is created
  - you are responsible for creating the table as well as the trigger and stored procedures
- trigger on/off option in RS 11.0

# Conclusions

- More and more people get interested in replication services
- In almost every presentation the replication server was mentioned
- Maybe Info Support must start investigating the replication server more seriously

In gebruik bij - IFF Tilburg
? Verdeer en verbersteat

# Engines, Tasks and the SQL Server Scheduler

Presentation to the Sybase European Users' Conference
9 October 1995

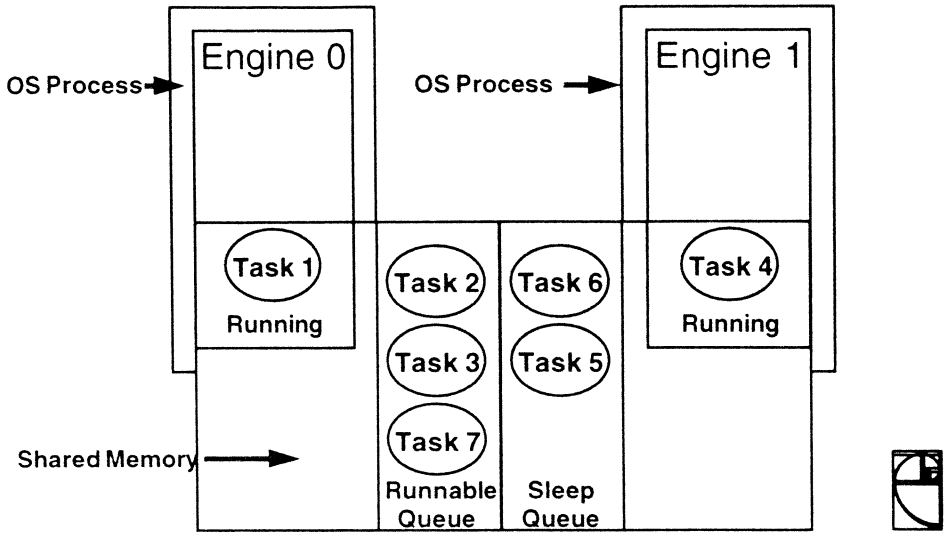

© Copyright 1995 Sybase™, Inc.

# Key Definitions

- Process
  - » An execution environment scheduled onto physical CPUs by the operating system
- Engine
  - » A process running the SQL Server binary which communicates with similar processes via shared memory. Together, these engines constitute a single server.
- Task
  - » An execution environment within SQL Server, scheduled onto engines by the SQL Server


© Copyright 1995 Sybase™, Inc.

# Engines and Tasks





© Copyright 1995 Sybase™, Inc.

# Engine Configuration Parameters

- **cmaxonline**
  - » Maximum number of online engines
  - » Server bootstrap will attempt to start this many engines.
  - » Should have at least as many physical CPUs
  - » Same as **sp_configure** parameter, **max online engines**
- **cminonline**
  - » Enforced, but currently of little use
- **cengadjinterval**
  - » Unused


© Copyright 1995 Sybase™, Inc.

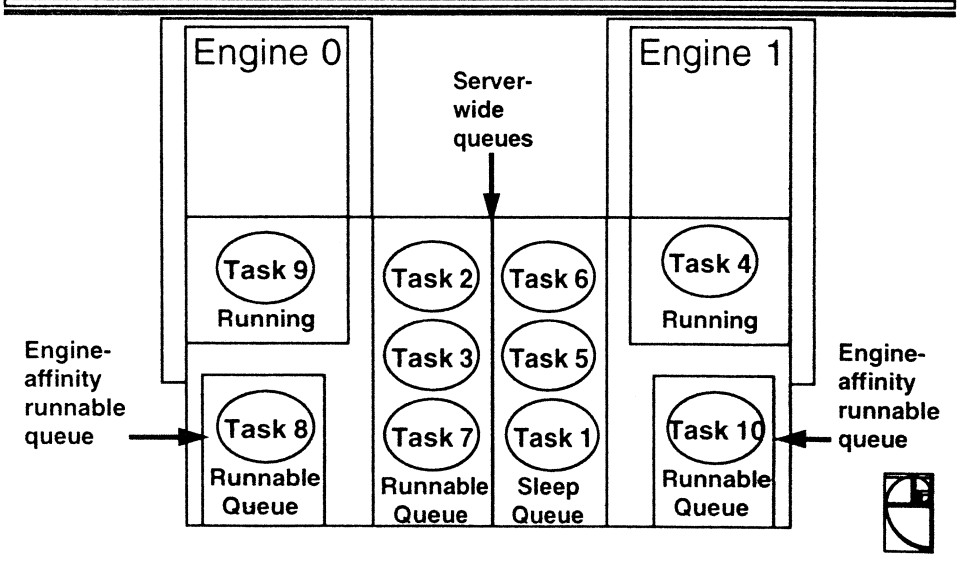
# Engine Affinity

- A task which must be run on a particular engine is said to be ***affinitied*** to that engine
- A task which is not currently affinitied can run on any engine
- Some system (kernel) tasks are affinitied throughout the life of SQL Server
- User tasks may be affinitied for short periods



© Copyright 1995 Sybase™, Inc.

# Engines and Tasks with Affinity

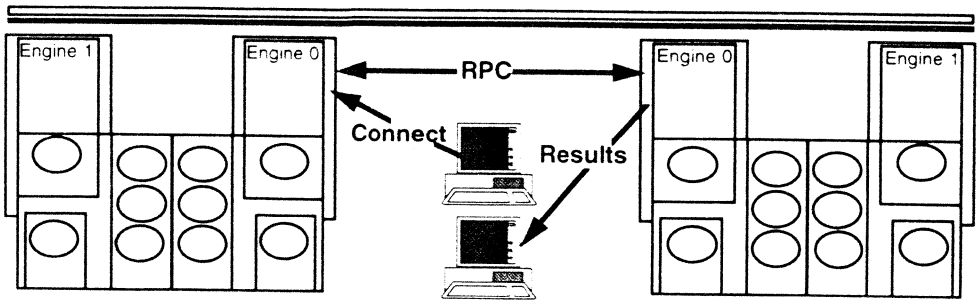







© Copyright 1995 Sybase™, Inc.

# Network Affinity in System 10




- All network I/O is done on engine 0
  - » Network listener task which handles user connections
  - » Network I/O completion task
  - » User tasks are temporarily affinitied when starting network I/O


© Copyright 1995 Sybase™, Inc.

# Problems with Network Affinity

- Single engine network I/O can create a bottleneck in multi-engine servers
  - » A single engine performs all the network I/O, thereby overloading it
  - » Other engines have to wait for engine 0 to perform the I/O
- Scalability is reduced


© Copyright 1995 Sybase™, Inc.

# System 11 Network I/O

- System 11 introduces ***multiple network engines***
- Network I/O can be done on any engine
- Engine 0 still handles user connections
- One network I/O completion task per engine
- After connection, load balancing is used to select an engine for network I/O
- Reduces network I/O bottleneck
- Allows more network connections


© Copyright 1995 Sybase™, Inc.

BJ inloggen : user koppelt één engine en blijft daar.
Er blijven dus load balance problemen.

# Engine Synchronisation

- Since memory is shared between engines, engines must synchronise access to shared data structures
- Failure to do so can result in corruption
- Protection of critical sections is done using
  - » spinlocks - engine loops while waiting
  - » semaphores - task is rescheduled while waiting


© Copyright 1995 Sybase™, Inc.

# Spinlock Granularity

- One spinlock to protect all of shared memory
  - » Easy to implement!
  - » Poor performance - single threaded server
- One spinlock to protect a single occurrence of each shared data structure
- Need to strike a balance with spinlock granularity
- Granularity has been increased for ***hot*** spinlocks
  - » Buffer manager has per-cache spinlocks
  - » Lock manager uses per-hash table spinlocks



© Copyright 1995 Sybase™, Inc.

# Keeping Time Within the Server

- Each engine operates a clock which interrupts normal server processing on a regular basis
- This enables housekeeping functions to be performed on a regular basis
  - » updates monitoring and other statistics, eg. CPU time
  - » handle processing of alarms eg. **WAITFOR DELAY**
  - » checks currently running task for timeslice errors
- Time of day still handled by the O/S



© Copyright 1995 Sybase™, Inc.

# Cclkrate Configuration Parameter

13

- Frequency of this interrupt is governed by the **cclkrate** configuration parameter
  - » Units are in microseconds
  - » Granularity is O/S dependent
  - » Typical value is 100000 microseconds (1/10 second)
- Sometimes this is referred to as a server ***tick***



© Copyright 1995 Sybase™, Inc.

# SQL Server Tasks

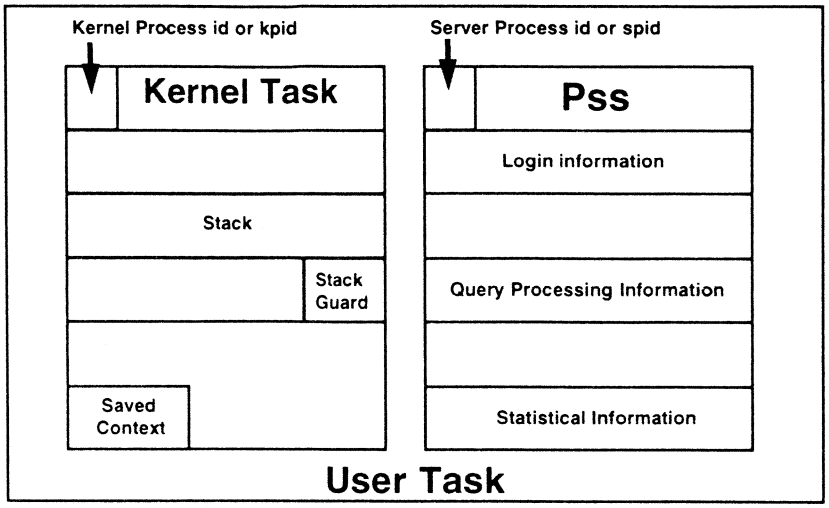
14

- A task, or ***thread*** is an entity that is schedulable by the server scheduler
- Analagous to an operating system process
- Managed ***within*** the server on most platforms
- Server bootstrap creates some tasks
  - » network listener task
  - » network I/O completion task
- User tasks are created when a client makes a connection to the server
  - » Limited by the parameter **cusrconnections**



© Copyright 1995 Sybase™, Inc.

# User Task





© Copyright 1995 Sybase™, Inc.

# Components of Kernel Task

- The kernel process id, or **kpid** is a unique number which the scheduler uses to manipulate the task
- Stack
  - » Size determined by **cfgstacksz** parameter
- Stack guard area
  - » Used to prevent stack overflow
  - » This has a signature that the server routinely checks for
  - » If the signature is corrupted, a stack overflow error occurs
  - » Size determined by **cguardsz**, typically 2048 bytes


© Copyright 1995 Sybase™, Inc.

# Pss

- Most tasks, including all user tasks, have a ***Process Status Structure***, or ***Pss***
- Sybase sometimes uses the word ***Pss*** to mean task
  - » For example, the message "Pss found with open sdes"
- All tasks with a Pss will be reported in the system table, **sysprocesses**
- The Pss contains most of the information required for query processing


© Copyright 1995 Sybase™, Inc.

# Infected Tasks

- A task which causes an unexpected exception is considered bad, or ***infected***
- There is no choice but to terminate the task
- Typical causes of infection are
  - » signal 10, a bus error resulting from unaligned access to data
  - » signal 11, a segmentation violation from dereferencing an invalid address
- Infection while holding a spinlock results in a server crash


© Copyright 1995 Sybase™, Inc.

# Scheduler

- ***Good guy*** scheduler
  - » A task yields control to the scheduler at convenient points
  - » Not pre-empted
- One scheduler task per engine
- Kernel task
  - » No Pss
  - » Does not show up in **sysprocesses**


© Copyright 1995 Sybase™, Inc.

# Yielding Task

- A running task will voluntarily yield when it has used up ***at least*** its timeslice
- A task will automatically yield when it goes into a sleep state
  - » waiting for I/O to complete
  - » waiting for a lock
- A task that goes into a sleep state will therefore not necessarily run for its whole timeslice


© Copyright 1995 Sybase™, Inc.

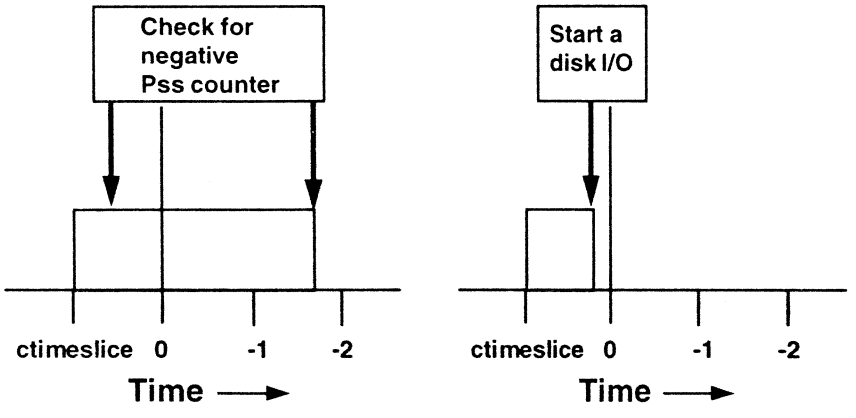
# Task Timeslice

- The configuration parameter **ctimeslice** defines the size of the timeslice, in ***ticks***
- When a task is scheduled to run, a counter in the Pss is initialized to **ctimeslice**
- Each clock interrupt, this value is decremented
- At frequent intervals, the task checks whether the counter has gone negative
- If so, it voluntarily gives up the CPU
- This scheme means that a task can run for more than **ctimeslice** ticks



© Copyright 1995 Sybase™, Inc.

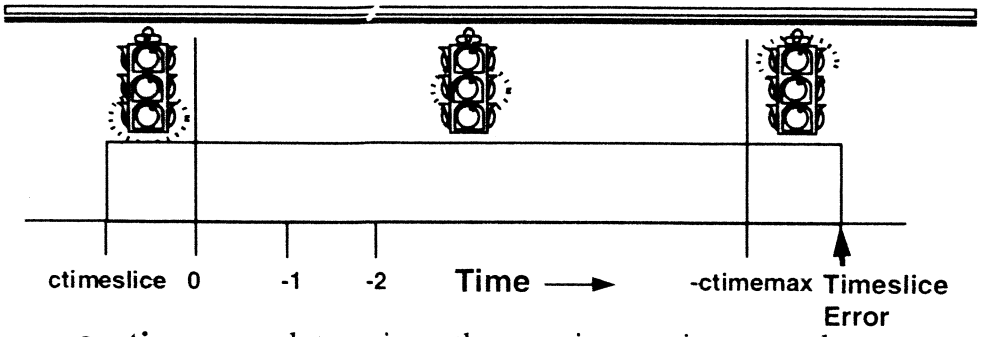
# Task Yielding to Scheduler






© Copyright 1995 Sybase™, Inc.

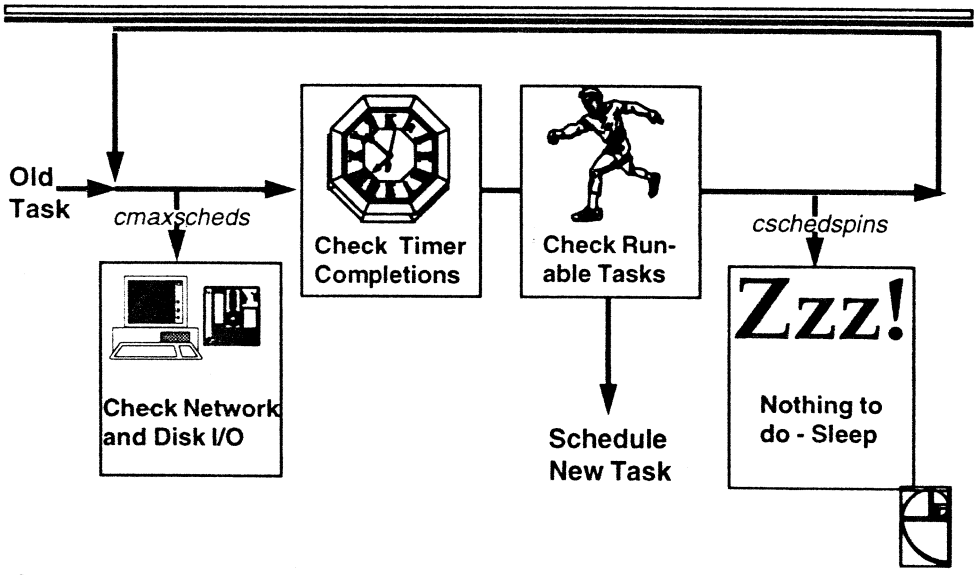
# Timeslice Errors




- **ctimemax** determines the maximum time a task can run without yielding
- The Pss counter is checked each clock interrupt for a value less than -(**ctimemax**)
- If so, it is infected as having a timeslice error


© Copyright 1995 Sybase™, Inc.

# Scheduler Task

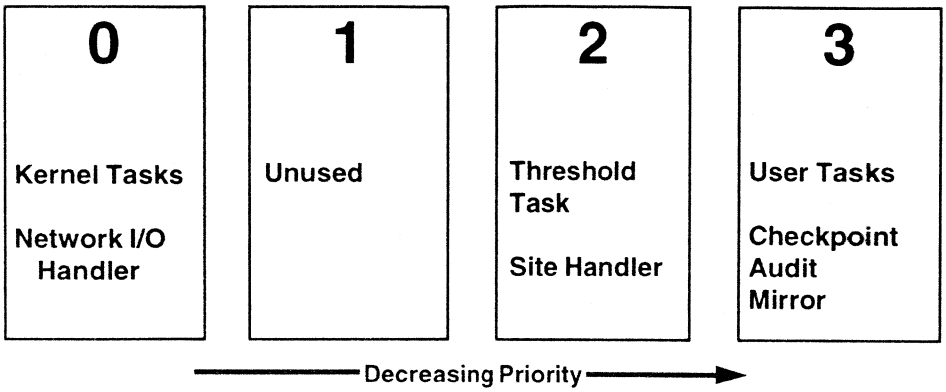




© Copyright 1995 Sybase™, Inc.

Een timeslice = 100 msec
Binnen de timeslice kunnen vele kleine processen draaien.
cmaxscheds bepaald de snelheid van het testen op de
timeslice.

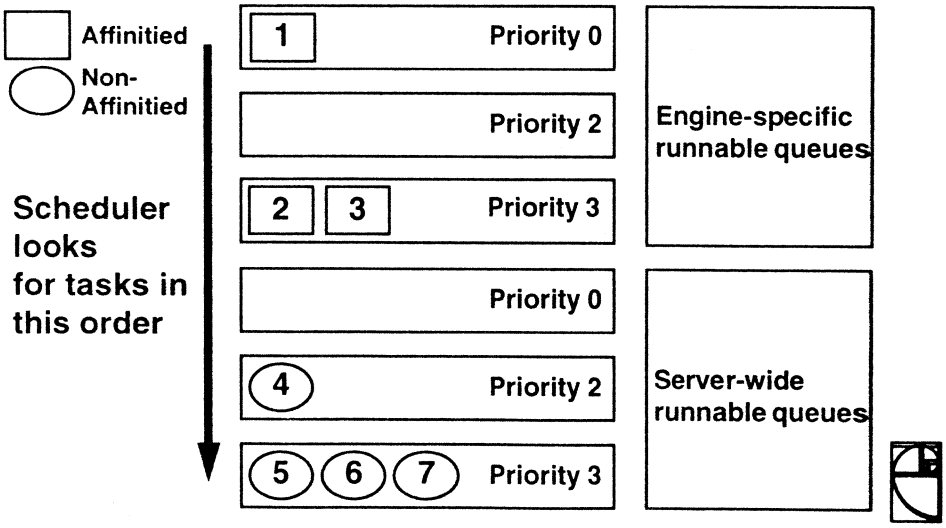
# Task Priorities




- Cannot change the priorities of tasks


© Copyright 1995 Sybase™, Inc.

# Selection of Next Runnable Task





© Copyright 1995 Sybase™, Inc.

# Server Idling

- When there is nothing to do, the scheduler enters an idle loop
- Repeatedly checks for work to be done
- The number of times this is done is given by the configuration parameter **cschedspins**
- Has the effect of using CPU, even though server is idle
- If no work is found after **cschedspins** iterations, the scheduler sleeps for a short time



© Copyright 1995 Sybase™, Inc.

# Responsiveness to I/O

- On many platforms, a busy scheduler will only check for I/O events only once in every **cmaxscheds** times it schedules a runnable task.
- The **cmaxscheds** configuration parameter is therefore a means of balancing I/O-bound tasks against CPU-bound
- Defaults to 10



© Copyright 1995 Sybase™, Inc.

# Tuning

- Best results obtained by application and database design
- Default values are often sufficient
- Changes should be made one at a time
  - » Measure effect of change
  - » Backout the change if it has a negative effect
- Do this with Sybase Technical Support supervision, ***or do it at your own risk***


© Copyright 1995 Sybase™, Inc.

# Changing Configuration Parameters

- Use **sp_configure** for ***user*** parameters
- Use **buildmaster** for ***internal*** parameters
  - » **buildmaster -d***masterdevice* **-y***parameter=value*
- Parameters referred to are **buildmaster** parameters
- In System 11, all **buildmaster** parameters will be modifiable via **sp_configure**

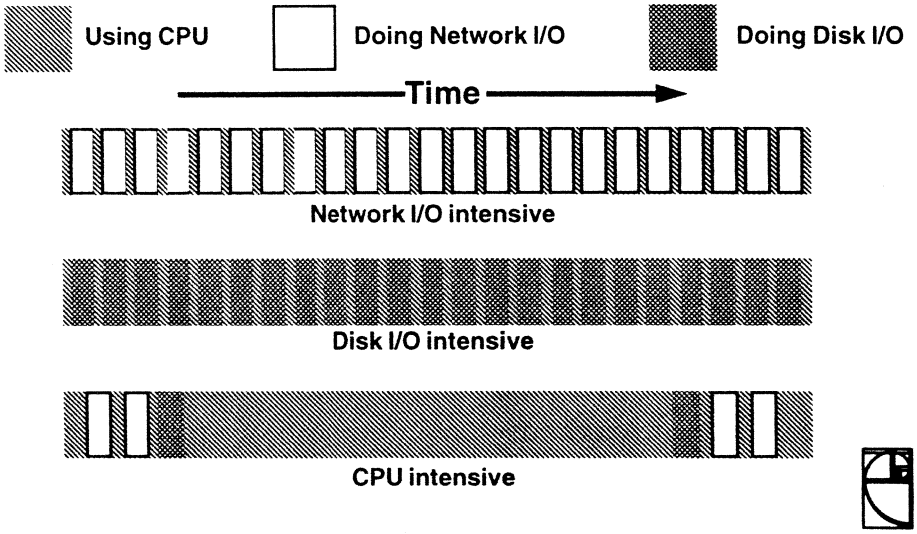

© Copyright 1995 Sybase™, Inc.

# Responsiveness

- The perceived time it takes to perform a job, such as a transaction
- Depends on the characteristics of the job
  - » CPU-bound
  - » Network I/O-bound
  - » Disk I/O-bound


© Copyright 1995 Sybase™, Inc.

# Determining Load Mix







© Copyright 1995 Sybase™, Inc.

# Throughput



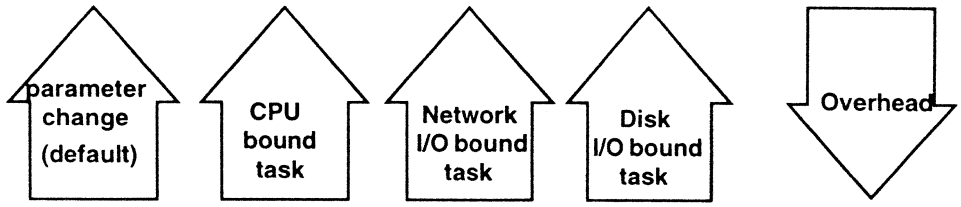
- The amount of work that the server is able to do
- Decreased throughput can be the result of extra overhead
  - » More task switching by the scheduler
  - » Checking for non-existent events
- Throughput will naturally increase if no trade-off is needed for gains in responsiveness
- Therefore use increase or decrease in ***overhead*** as a responsiveness-independent means of measuring throughput changes.



© Copyright 1995 Sybase™, Inc.

# Responsiveness and Throughput






**Default/ Recommended value in brackets**

**Ideal: Increase in responsiveness**

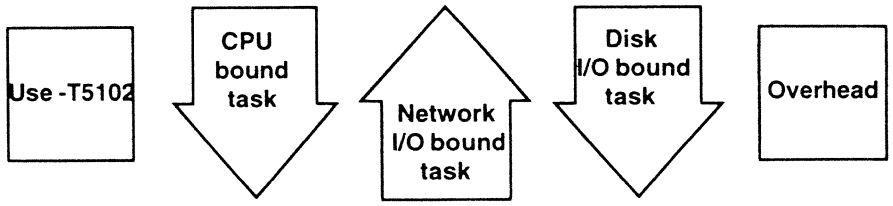
**Ideal: Decrease in overhead Increase in throughput**

- Increase in overhead does not always mean decreased throughput
  - » Increase in responsiveness might make up for the loss



© Copyright 1995 Sybase™, Inc.

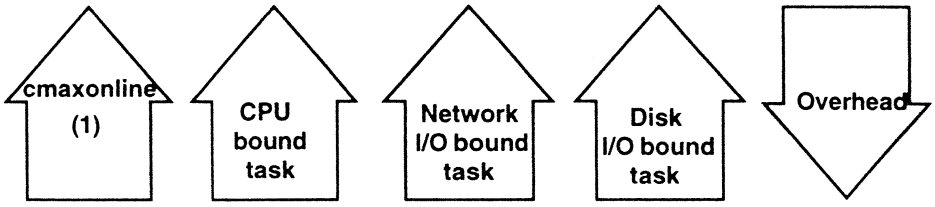
# Taking Advantage of Task Affinity




- Trace flag 5102 ensures that engine 0 only runs affinitied tasks
  - » Has the effect of dedicating engine 0 to network I/O
  - » Avoids CPU intensive tasks running on engine 0
  - » Need at least 4 engines online
- This can improve performance on servers which perform large amounts of network I/O


© Copyright 1995 Sybase™, Inc.

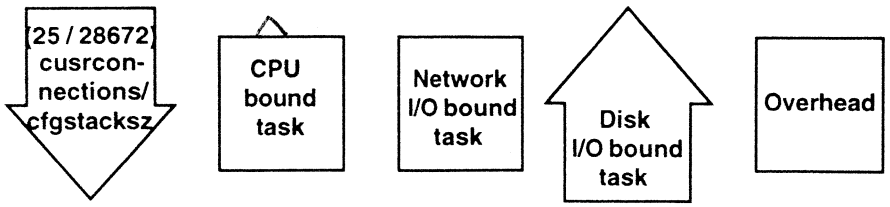
# Cmaxonline




- Should only be increased to a value equal to the number of processors
- More engines available for processing
- May want to leave one processor for other processes
- Better responsiveness and throughput


© Copyright 1995 Sybase™, Inc.

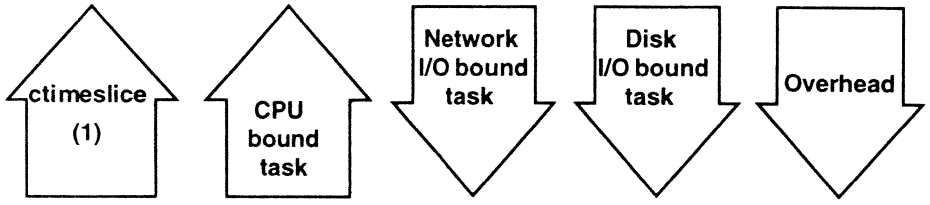
# Cusrconnections/Cfgstacksz




- On servers with little memory, decreasing the number of user tasks (**cusrconnections**) and stack size (**cfgstacksz**) can free up memory for the buffer (or procedure) cache
- A single user connection costs around 32K with **cfgstacksize** set to 28K


© Copyright 1995 Sybase™, Inc.

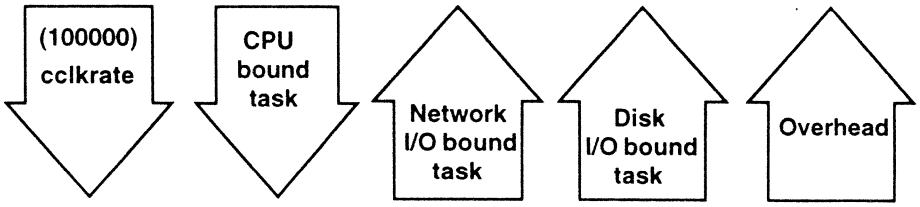
# Ctimeslice




- Allows CPU-bound tasks to run for longer before scheduling out
- Less task switching and more stable buffer cache mean decreased overhead
- Major degradation to I/O responsiveness
- Not recommended


© Copyright 1995 Sybase™, Inc.

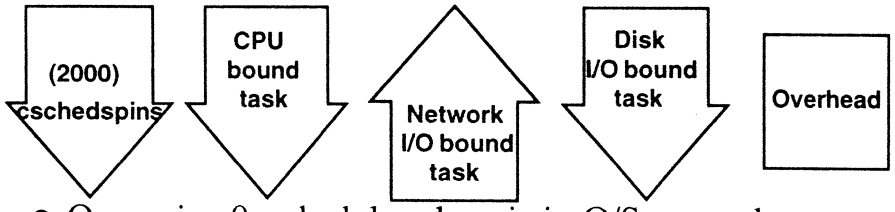
# Cclkrate




- Only means of emulating a ***decrease*** in **ctimeslice**
- CPU-bound tasks run for a shorter time
- More evenly balanced with I/O-bound tasks
- Too much decrease results in high interrupt overhead. Do not decrease beyond 10% of default


© Copyright 1995 Sybase™, Inc.

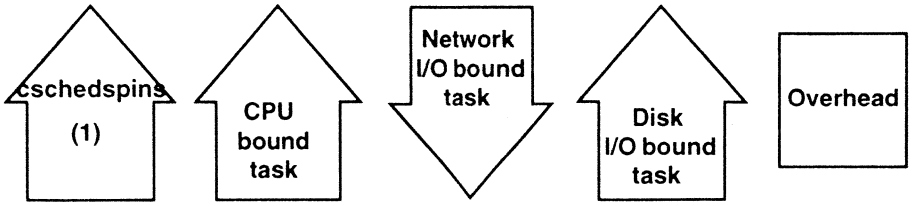
# Cschedspins (Multi-Engine)




- On engine 0, scheduler sleep is in O/S network poll routine
- Engine 0 will be more responsive to the network
- All engines will be less responsive to CPU-bound tasks, and disk I/O completion
- O/S needs to re-schedule process more frequently


© Copyright 1995 Sybase™, Inc.

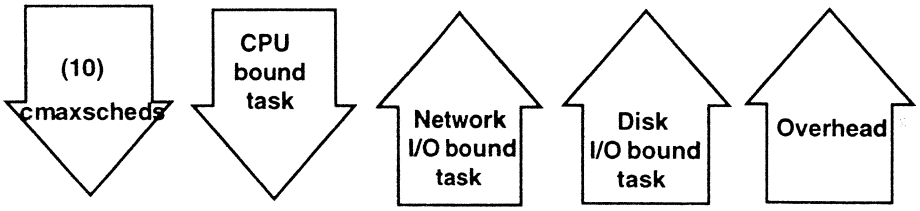
# Cschedspins (Single-Engine)




- Responsiveness to network events will decrease
- Responsiveness to disk events and CPU-bound tasks will increase
- Competing processes can be starved of the CPU on single processor machines
- Value between 1 and 10 is recommended


© Copyright 1995 Sybase™, Inc.

# Cmaxscheds




- Increases the number of times that I/O events are checked for
- Therefore responsiveness increased for I/O bound tasks
- Overhead of sometimes checking for non-existent I/O events


© Copyright 1995 Sybase™, Inc.

## Goals of SQL Server 11.0

✓ ***As always:***

**Deliver the Maximum Power of the Underlying System to the User Application.**

## SQL Server 11.0 Themes

### ✓ Decision Support & Mixed Workload

- **Workload Adaptable**
- **Speed-up Data Load (BCP) Operation**
- **Speed-up Data Scans**
- **Better Optimization of Large/Complex Joins**
- **Improved Subquery Processing**
- **Prevent DSS from Impacting OLTP Resources**

# SQL Server 11.0 Themes

## ✓ SMP Scalability

- Symmetric Kernel
- Distribute Work to each CPU
- Balance Workload of each CPU
- Prevent CPUs from Contending for Common resources

11.0 limit to processors

# SQL Server 11.0 Themes

✓ Improved Concurrency

- Isolation level 0 ("Dirty Reads")
- Locks Aligned with Engines (for SMP)
- Almost all Updates "In-place"
- Specify "max_rows_per_page"
- Configurable "lock promotion threshold"
- Page-split Reduction
- Tunable "dead-lock detection frequency"
- Lock-wait Monitoring

# SQL Server 11.0 Themes

## ✓ Efficient Memory Management

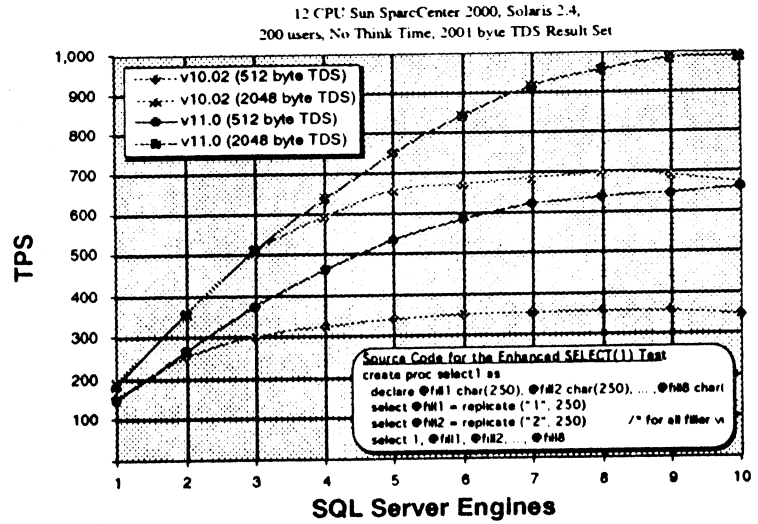
- Reduce Physical I/O (Improve Cache-hit Ratio)
- Cater to Special Memory Needs of Applications
- Manage Memory Effectively
- Reduce Data Access Contention

# SQL Server 11.0 Themes

## ✓ Easy Install/Upgrade/Configuration

- Simplified SYBINIT
- Dump Compatibility with System 10 Dumps
- New Hierarchical Configuration
- Readable/Editable Configuration File
- Settable Attributes for Objects
- Database on-line/off-line Commands

# Multiple Network Engines in v11

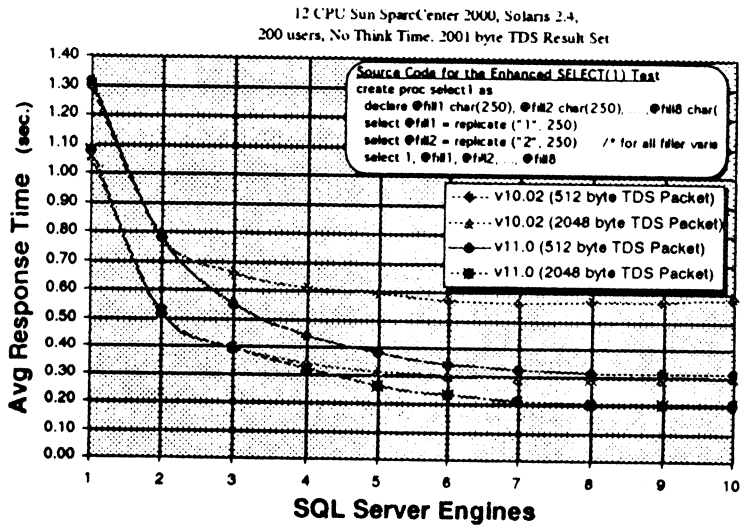




- v10 performed networking only on Engine 0 in SMP configurations increasing contention and resulting in decreased throughput.
- v11 allows all engines to perform networking operations increasing throughput.
- In network-intensive apps (large result sets, text/image,etc), consider additional engines to help increase throughput.
- Don't overlook the benefit of larger TDS packets before buying that next CPU!



Copyright Sybase Inc. 1990-1995, All rights reserved

# Multiple Network Engines in v11



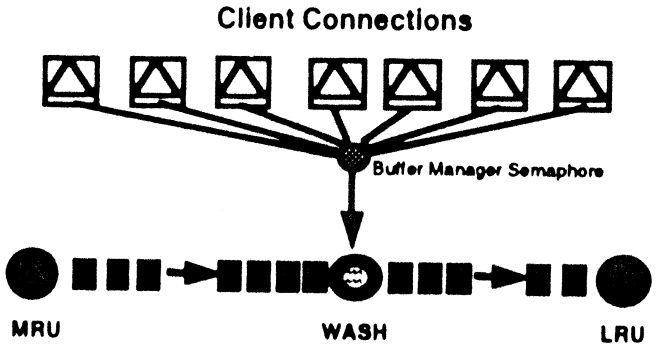

- At the same time throughput increases from MNE, response time decreases.
- In network-intensive apps, consider additional engines to help
- Don't overlook the benefit of larger TDS packets before buying that next CPU!
- Monitor with *sp_sysmon* to show network load by engine. Look at the distribution of clients to network engines with *dbcc engine(net, showall)*

Copyright © Sybase, Inc. 1988-1998. All rights reserved.

## Memory Manager

- In System 10, Single Data Cache
- In SQL Server 11.0:
  - Multiple Named Data Caches
  - Large I/O Bufferpools
  - New Buffer Replacement Strategy
  - Integrated Optimization
  - Background Housekeeper Process

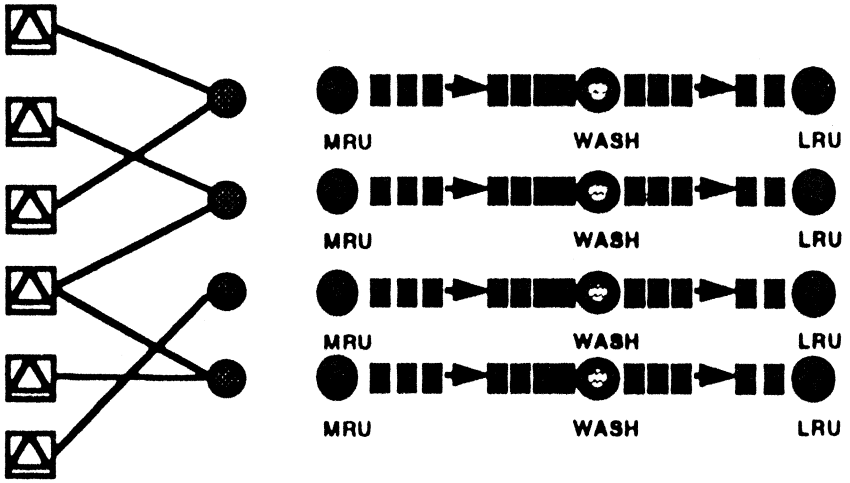
# Single Cache in SQL Server 10.x



# SQL Server 11.0: Named Data Caches

- User Created Data Caches
- Restrict Cache to Specific Objects/Appls.
  - On-line bind or unbind objects to a cache
- Each Cache Individually Configurable
- Better Memory Utilization
- Protect "hot" Objects from Page-Faults
- Reduced Contention on SMP Platforms

# SQL Server 11: Multiple Named Caches

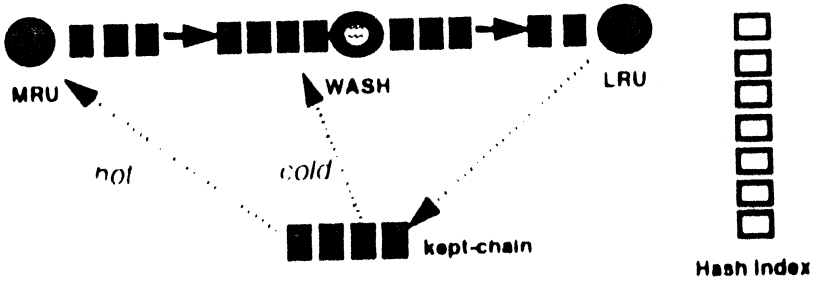


# Cache Strategy Optimization

- No Changes in Application Required
  - Caching Strategy Generated by the Optimizer
- New Buffer Aging Mechanism
  - Buffers Designated "Hot" or "Cold"
  - Different Scheme for Index and Data Pages
  - Improve Cache-Hit Ratio
- Sequential Prefetch for Large Scans
- Statistical Model for Caching Analysis
- SHOWPLAN Displays the Strategy

# Cache Strategy Optimization

- Better Tuning/Troubleshooting Options
- For a given Configuration, Optimizer Transparently Picks Strategy
  - Whether to do a Sequential Prefetch
  - Whether to use fetch-and-discard strategy






Hash Index

# Using Multiple Buffer Caches in v11


## Minimize Physical I/Os

- Lock "Hot" tables in cache by creating cache large enough to hold entire table. Optimizer assumes cache will be primed by queries doing "normal" caching. Cache can be manually primed by forcing table scan on data pages. For example,

  *SELECT count(*) from table(0)* ***prefetch 16K***

- Configure both 2kb and 16kb Pools to enable large I/Os.

- SQL Monitor Client will provide Cache Analysis features to show recommended I/O sizes by object (tool may only be beta by v11.0 GA).

- *sp_sysmon* will provide "system-level" information on:
  - ☑ Pool Utilization by Cache (% of logical I/Os by Pool by Cache)
  - ☑ % of Large I/O Requests Denied
  - ☑ "Page chain" Fragmentation ? (Avg Pages Used per Lrg I/O).
  - ☑ Percentage of Buffers Following "Fetch/Discard" Strategy.

# Using Multiple Buffer Caches in v11


## Minimize Physical I/Os

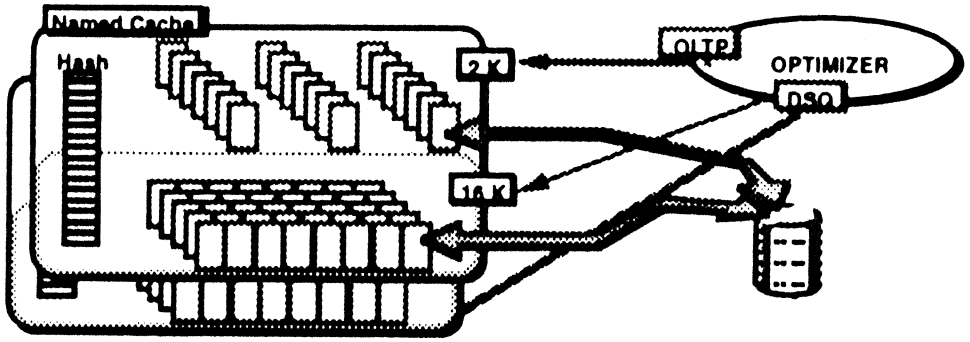
- Without it, refer to the Access Path Analysis done as part of your Data Modeling effort and follow these general guidelines *(**NOTE:** these may change before GA)*:
  - ☑ Fast BCP and DBCC 3-5 times faster with Large I/O.
  - ☑ Large I/O strategy generally only chosen by optimizer on Table Scans and Clustered Index & covered queries access when more than 3 pages costed.
  - ☑ Large joins via clustered index is now more a much more interesting alternative choice of which index to cluster.
  - ☑ If query too slow due to physical I/O, look at cache strategy! Is the optimizer incorrectly choosing "Fetch & Discard" on important objects. Strategy used on work tables and when scan will use MORE than 50% of the pool's total buffers.

# Large I/Os

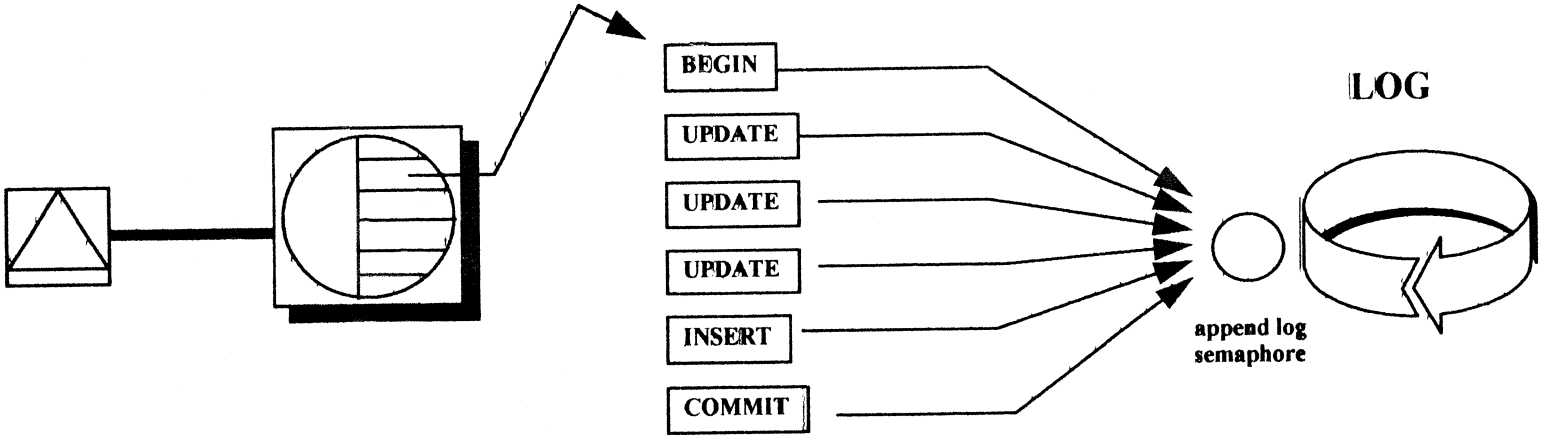
- Utilize Larger Block-size
- Multiple Buffer Pools:
  - For every Named Cache
  - On-line Reconfigure
- Provide Sequential Prefetch
  - Many-fold Increase in Scan Performance
- Several-fold Faster Bulk Loads (BCP)
- Variable-size I/Os: 2K, 4K, 8K & 16K
  - In some cases up to 512K

# Workload Adaptable

- Housekeeper Task Off-loads I/Os from TP
- Handle Mixed-workload
  - DSS Queries do Large Block I/Os
  - OLTP incurs 2K I/Os



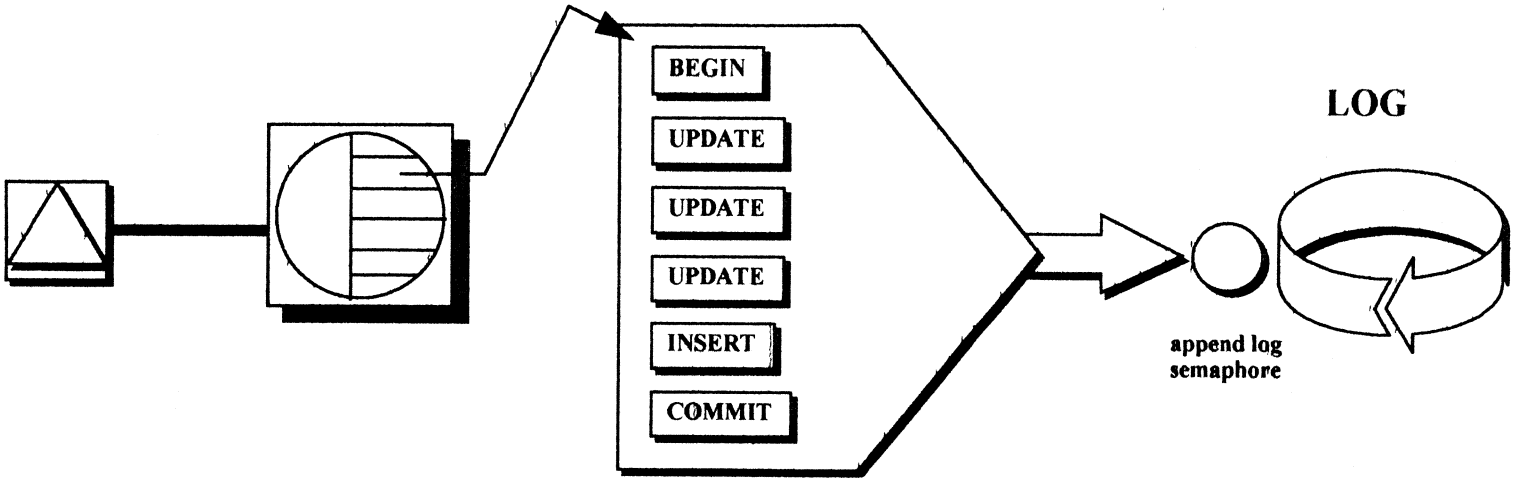
# Current Logging System




- Each Log Record is Transferred One-by-One to Log
- Each Transfer goes Through the Database Wide Log Semaphore

Sybase Proprietary and Confidential

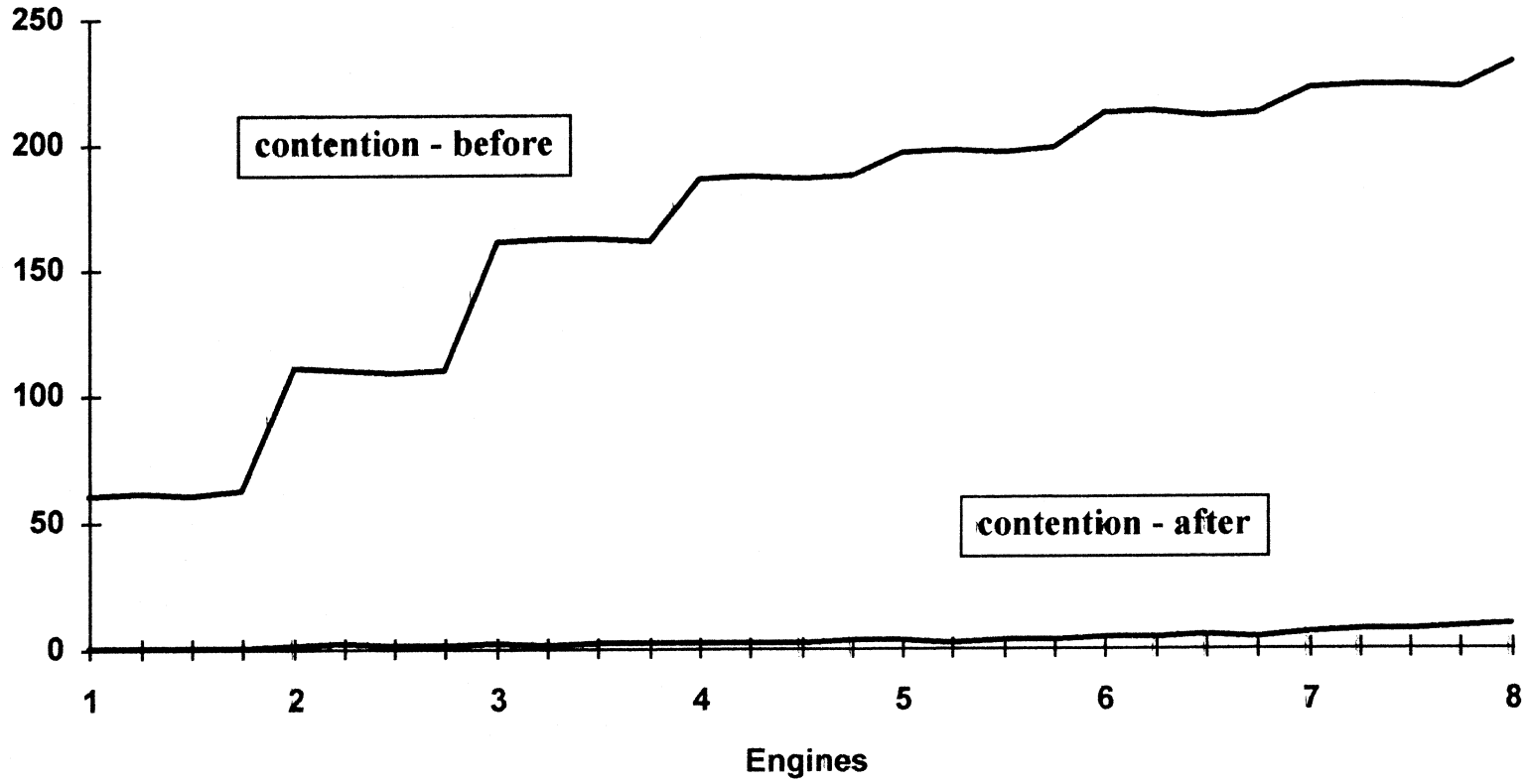
# Private Log Cache




- Records are Transferred in Batches to the Log
- Reduces Contention on the Log Semaphore by a Large Factor

Sybase Proprietary and Confidential

# Release 11 Log Performance




Sybase Proprietary and Confidential

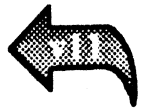
# Tuning Transaction Logging in v11


## Reduce SMP Contention and Minimize I/O

- Proper Sizing Important! Memory is per-user so sizing too big in relation to Xact size is wasteful. Alternatively, too small forces more physical I/O! In general, choose a size based on average transaction size!
  - Tune with *sp_sysmon* via :
    - ☑ Maximum Transaction Size (use input to size PLC)
    - ☑ % of PLCs Flushed Full
- Number of PLCs per PLC spinlock (for SMP only) allows additional tuning for SMP contention.
  - Tune with *sp_sysmon* via :
    - ☑ PLC Contention (higher numbers indicate more PLC spinlocks useful)
- Physical I/O sizing of Transaction Log is important. Bigger is not always better! Trade-off is Group Commit efficiency (better throughput by minimizing physical I/O) vs. longer response times waiting for COMMIT I/O. Use sp_logiosize to tune the size of the log buffer. Make sure to bind the log to its own cache.


Copyright © 1992 by Sybase, Inc. All rights reserved.

# Tuning Transaction Logging in v11




## Monitoring Oldest Active Transactions

- Recoverability Prevents Log Truncation Past the Last Open Transaction
- Causes Significant Operational Problems in v10 and Below Because of No Reasonable Method to Find and Deal with the Offending Transaction.

☑ v11 Adds the *syslogshold* Virtual Table to Hold Information about each Databases' Oldest Active Transaction and Rep Server's Truncation Point.

☑ DBAs can run the following command to see if Log Truncation is Blocked.

```
SELECT   H.spid, H.name,
FROM     master..syslogshold H, sysindexes I
WHERE    H.spid = db_id() AND H.spid != 0 AND I.id = 8 /* syslogs? */ AND
         H.page = I.first   /* First Page of Log = Page of Oldest Xact ? */
```

☑ DBAs can run the following command to Find the Oldest Transaction.

```
SELECT   H.spid,U.name, P.cmd, H.name, H.starttime, P.hostname,
         P.hostprocess, P.program_name,
FROM     master..syslogshold H, master..sysprocesses P, master..sysusers U
WHERE    H.spid = P.spid AND P.suid = U.suid AND
         H.spid != 0                  /* Not Replication Truncation Point */
```

☑ If so, Use the Kill Command to Terminate the Transaction or Insert Logic into the "Last Chance Threshold" to abort the Transaction Before Log Fills.



Copyright © Sybase, Inc. 1984-1998. All rights reserved.

# Data Partitioning Issues

- Syntax:
  - alter table tabX partition n
  - alter table unpartition
- Some Operations Disallowed
  - Clustered Index, Truncate Table
- New syspartitions catalog
- When Use?
  - If lot of Insert-contention, e.g.. History, Audit Table
  - Partition, Load Data in Multiple Sessions, then Unpartition

## Data Partitioning Issues

- For Insert-intensive Application
  - Reduce Data Insert Contention
  - No Index Overhead
- For Spreading Data across Devices
  - Round-Robin per Transaction
  - Data on all Devices of Data Segment
  - Applications may use over 100 Partitions
- Useful for Faster BCP Loads
  - Recent Tests Show over 6 GBytes/Hr.

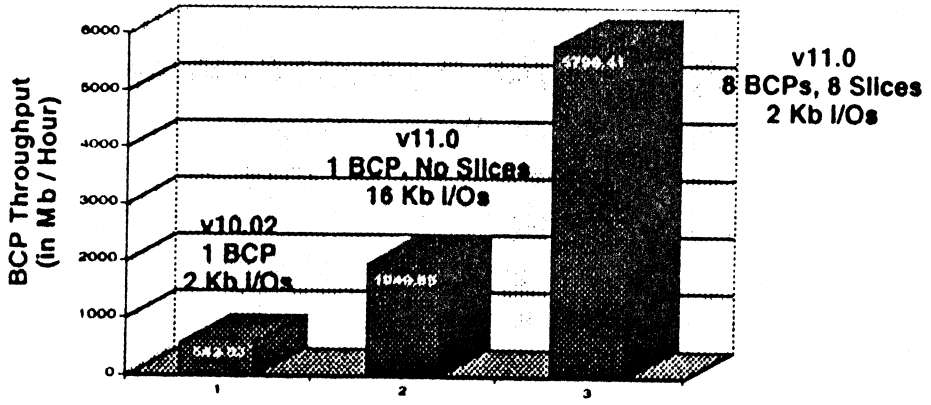
# Tuning SQL Server 11.0 for BCP

## Large I/Os

- Removes a Major Performance Bottleneck to BCP
- *Always* Do Fast BCP with Large I/Os (up to 16 kb, 256kb Possible in the Future)
- Today, Don't Yet See Any Real Benefit With Slow BCP

## Table Slices

- Allows Load of Single Table to be "Manually" Parallelized
- Fastest Method of Loading Data, Especially in Conjunction with Large I/Os
- If Clustered Index Needed, Partition For Load, then Unpartition (via Alter Table)



# Tuning Space Allocation / Deallocation in v11

## Pre-v11 Algorithms Degraded Performance As Tables Grew

- Significant Reduction in BCP Speeds:

| Rows | Pages | Rows per Second through fast bcp |
|---|---|---|
| Start | < 1000 | 3,360 |
| 15,299,575 | 527,570 | 2,851 |
| 43,348,980 | 1,494,800 | 2,229 |
| 162,263,198 | 5,595,282 | 942 |
| 368,164,062 | 12,695,312 | 349 |

- Due to multiple OAM scans (2-5) on longer & longer OAM chains. Affects all operations doing space allocation & deallocation

## v11 Algorithms Rely on "Hints" Stored on OAMs


- Suggests where to "likely" find unused pages.
- Hints created automatically as pages allocated / deallocated
- Tuning revolves around optimizing when to search for a free page or just allocate another extent. Use *sp_configure "OAM Scan Threshold", #*
- Monitor with *sp_sysmon* to configure properly.


Copyright Sybase, Inc. 1984-1994. All rights reserved.

# Dirty Reads in v10.1 & 11


## Run Queries without Locks When Semantically Possible

- ☑ Dirty Read queries acquire NO database locks. Therefore, they can NOT block other queries, nor can other queries block them!
- ☑ Dirty Reads allow reading uncommitted data. To use, applications MUST be tolerant of incorrect / missing data

  ✓ **Examples of *Good* Uses for Dirty Read Queries:**
  - ➡ Count of Rows About to be Processed
  - ➡ Estimate of Percent of Sales by Region for Market Forecasting

  ✗ **Examples of *Bad* Uses for Dirty Read Queries:**
  - ➡ Account Balance Process for Financial Institutions
  - ➡ Airport Runway Scheduling

- ☑ ***Syntax***: set transaction isolation level {0 | read uncommitted} ** session level
  select ... at isolation level {0 | read uncommitted} ** statement levels:
  declare cs1 cursor for select ... at isolation level {0 | read uncommitted}
  readtext ... at isolation level {0 | read uncommitted}


Copyright © 1994-1997 All rights reserved

# Dirty Reads in v10.1 & 11


## Things To Watch Out For Using Dirty Reads

### ✗ Requires UNIQUE Index be Chosen

- Like cursors, scans must be "restarted" if rows move underneath it.
- Server retains index key values and re-issues select after manipulating the where clause in order to find the row. Significant re-starts will slow processing.

| a | b | c |
|---|---|---|
| 1 | 1 | 2 |
| 1 | 2 | 1 |
| 1 | 2 | 2 |
| 1 | 3 | 1 |

*Current Scan Position: Re-start @ WHERE a=1 AND b=2 AND c=1*
*If Not Found, Try @ WHERE a=1 AND b=2 AND c >1*
*If Still Not Found, Try @ WHERE a=1 AND b>2*

### ✗ Huge Risk if Combined With Forceindex of Non-unique Index

- Rows Re-read or Skipped Without Error
- Infinite Loop Possible if Volatile Table

### ✗ Explicitly Set Isolation Level in SPROCS for *Consistent* Behavior

- If set outside sproc, retains session level for current user.
- Will cause recompilation if sproc created at level 1 and next user's level is 0.
- Retains unique index selection in plan even if level changes to 1 execution

Copyright © Sybase, Inc. 1994-1998. All rights reserved.

# Improved Concurrency: Dirty Reads

- Isolation Level 0 ("Dirty Reads")
  - Doesn't Acquire Any Locks
  - Doesn't Block Writers
  - May Read "uncommited " Data
- Session/Query Level Options
  - Reads at Isolation Level 0
  - But Writes at Isolation Level 1 or Above
- Requires Unique Index
  - Option to Automatically Generate Unique Index

# Improved Concurrency: Dirty Reads

- Syntax:
  - Select ... at isolation read uncommited
  - set isolation level 0
- Can Change Isolation Level with Session
  - Implications for Stored Procedures
  - Implications for Cursors
- Automatic Identity for Unique Index
  - sp_dboption db1, "identity in nonunique index", true
  - sp_dboption db2, "auto identity", true
  - Used only by Dirty Reads (updatable cursors at level 0)

# v11 Changes Affecting Concurrency

## Configurable Lock Promotion

v10.1 & 11

- ☒ At 200 pages of acquired/held page locks, SQL Server attempts, on every subsequent lock request, to promote all page locks to 1 table lock! ***Very*** problematic for VLDB (eg., 200 pages locked on a million row only 0.1%)!
- ☑ v10.1 adds a server-wide configuration option (via sp_configure "lock promote") A start, although will have problems in environments with tables of mixed sizes.
- ☑ v11 provides dynamically configurable lock promotion at the server-wide, per-database, or per-table levels. Percentage-based and low & high water mark specification allows growth flexibility without losing control of resource consumption for *real* VLDB and mixed workload environments!
- ☑ ***Syntax:*** sp_lockpromote {PCT | HWM | LWM},
  <value>, {SERVER | DATABASE | TABLE}, <db or table name>
- ☑ Lock Promotion Threshold (LPT) = (PCT * table_size) / 100. If LPT < LWM locks promoted at LWM. IF LPT > HWM, locks promoted at HWM!
- ☒ ***Gotchas***: Easy to run out of locks if not careful

Copyright Sybase Inc 1994-1998 All rights reserved

# Lock Monitoring and Configuration

- Configurable Lock Options
  - Isolation-level
  - Lock Promotion, using sp_lockpromote
    - Server-wide and Per-object as Percentage
  - max_rows_per_page
- Lock-wait Monitoring
  - Report Time SQL Server Spends Waiting for Various Locks
  - Per Object and Per Process Basis:
    - dbcc object_stats() and dbcc user_stats()

# Improved Concurrency: SMP

- **Improved SMP Locking**
  - Lock-wait Chains Per Engine Basis
- **Deadlock Detection Frequency**
  - System 10: Every Task Blocked by Lock Request Triggered Detection
  - SQL Server 11.0: Deadlock Detection Every 500ms
  - SQL Server 11.0: Frequency is Tunable
  - Tune According to Response Time Requirements

# v11 Changes Affecting Concurrency

## Persistent Fillfactor *(max rows per page)*


- Until Row Level Locking (RLL), tables and indexes can be forced to contain fewer rows. Unlike fillfactor, SQL Server guarantees the table or index will never contain more than X rows!
- Restricts the number of rows on:
  - ☑ Data Pages
  - ☑ Clustered Index *Leaf* Pages
  - ☑ Nonclustered Index *Leaf* Pages
- Obviously, this forces the table or index to occupy more disk space.
- Consequently, only recommended for critical tables identified as problems!
- Use "with max_rows_per_page = *X*" clause (*X* = 0-256)with CREATE TABLE, ALTER TABLE, CREATE INDEX to specify on Object by Object basis
- Default value of 0 forces current behavior (full leaf pages of both clustered & nonclustered indexes). More lock contention & page splits possible.
- **Gotcha's:**
  - ☒ "SELECT INTO" does not inherit value from base table. Must explicitly change it with *sp_chgattribute*. Might be a problem for temp tables!

# Update Processing

- System10: Most Updates = Delete + Insert
  - All Indexes Affected
  - More Chances of Contention/Deadlock
  - Twice the Amount of Log Records
- SQL Server 11.0:
  - Most Updates are In-place
  - Only Affected (unsafe) Indexes Updated
  - Special Optimization for Deferred Updates
  - More Cases of Deletes Also In Direct Mode
  - Requires Less Log Space
  - Mode Displayed by SHOWPLAN

# Update Rules in v11


**"Direct" Updates In Place**

- Modifications made immediately without *any* data movement on page
- Fastest Update Possible
- Done only when fixed length columns change
- "Unsafe" indexes *(key columns updated)* do DELETE / INSERT on index row(s) and make negate performance benefit

**"Cheap" Direct Update**

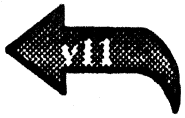
- Modified row fits on same page and put back in original place (same offset on page)
- Second Fastest Update Possible
- Done when variable length columns change or Update Trigger
- Before and After Image of Row is Logged
- "Unsafe" indexes *(key columns updated)* do DELETE / INSERT on index row(s) and make negate performance benefit

**"Expensive" Direct Update**

- Full DELETE / INSERT of Data and Index Rows
- Third Fastest Update Possible
- Done when row no longer fits on page or clustered index key(s) updated


Copyright © 1992 Sybase, Inc. All rights reserved

# Update Rules in v11




## "Deferred" Updates

- Now done whenever the follow conditions are met :
  - ☒ Conditions for "Direct" updates are not met
  - ☒ Updates using joins
  - ☒ Updates to columns used in Referential Integrity Checks

## v11 Optimizations to "Deferred" Updates

- Do as much as possible at run-time in Direct-mode
- Delete portion of deferred updates done in direct-mode
- Insert portion of deferred updates done in deferred-mode
- ☑ Requires only 1 pass back through log to process deferred inserts
- Example: UPDATE mytable SET col1 = "newvalue"
  - ☑ Row 1: Row fits - cheap direct update
  - ☑ Row 2: Row does not fit - direct delete, log deferred insert
  - ☑ Row 3: Because of row 2 delete, row 3 now fits - cheap direct update
  - ☑ When done, only if we did any deferred inserts logging do we do the log scan to process the deferred insert portion of the update.

Copyright © Sybase, Inc. 1984-1988. All rights reserved.

# Tuning Transact-SQL Queries

## Learn the Strengths and Weaknesses of the Optimizer

One of the largest factors determining performance is T-SQL! Test not only for efficient plans but also semantic correctness !

- ❑ **Optimizer "Plans" Join Order 4 Tables at a Time**
  - Every combination of "4-table permutations" is costed *(e.g. {t1,t2,t3,t4}, {t1,t2,t3,t5}, {t1,t2,t3,t6}, {t1,t2,t4,t5}, {t1,t2,t4,t6}, {t2,t1,t3,t4}, etc.).*
  - Best "outer" table is saved and the remaining combinations are "costed" to determine "next outer-most" table until done.
  - Scrutinize these types of queries for best possible plan.
  - Adding redundant predicates (eg. *where a = b and b = c and a = c*) gives the 4.9.1 and below optimizer more choices!
- ☑ **v11 Optimizer *Can* Plan Join Order *Up to* 8 Tables at a Time**
  - Designed for DSS although highly "normalized" OLTP will benefit too!
  - Optimization time **will** increase as the search space increases!
  - ***Syntax:*** *SET TABLE COUNT <N>  /* valid values for N = 1 - 8 */*
  - If optimization time too long for OLTP queries, use "force" methods instead.




Copyright © Sybase, Inc., 1994-1995. All rights reserved.

# Tuning Transact-SQL Queries

## Subquery Processing in SQL Server 11


- **Subqueries Processed "Outside-In"**
  - ✓ Evaluated once for every row in outer query.
  - ✓ In-memory subquery cache saves previous values! Good if duplicate join or correlation values. Especially good if values ordered or few distinct values. Memory borrowed from proc cache and dynamically resized as needed! View hit rate with *sp_sysmon* or run query with *set statistics subquerycache on*
    - ➧ select title_id from titles where price > ALL
      (select price from titles where advance < 15000)
  - ✓ "Short-circuiting" may prevent subquery evaluation by moving subquery to last step and stopping as soon as:
    - ➧ an AND clause = FALSE
    - ➧ an OR clause = TRUE
  - ✓ Eliminates sort step in DISTINCT expression subqueries *(ie column RELOP subquery).* Error occurs if subquery returns multiple rows.
  - ✓ Eliminates GROUP BY ALL step needed on v10
  - ✓ Correctness issues eliminated.
  - ✓ Subquery in OR clause executed as "nested loop", not join. Allows optimization.
- **SET "DUP_IN_SUBQUERY" is not needed now and is GONE!**
- **New Behavior Requires Drop & Recreate of SPROCs After Upgrade !**

Copyright © Sybase, Inc., 1994-1995. All rights reserved.

# Subquery Processing: SQL Server 11.0

- Many Correctness Problems Fixed
- VIEW Materialization Improvements
  - Queries with views & aggregates
- Some Restrictions Lifted
  - DISTINCT & GROUP BY allowed in same Subquery
- SHOWPLAN
  - Displays Subquery Execution Strategy
- Improved Optimization
  - Subqueries with OR Clause
  - Subquery/Join Ordering

# Subquery Processing: SQL Server 11.0

- Improve Performance of All Queries
  - Outside-in, Nested-loop Processing
- Execution Strategies:
  - Better Unnesting Model
  - Subquery Caching
  - Materialization
  - Short Circuiting
- Backward Compatibility
  - Need to Drop and Recreate Stored Procedures
  - Most Trace & Other Options Eliminated

## SQL Server 11.0: Subquery Processing

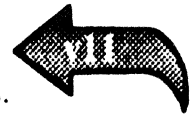
- Most Applications will be Affected
- Fixed Many Correctness Problems
- Fixed Many Performance Problems
- No Backward Compatibility Though
- Lift Many Coding Restrictions
- Enhanced VIEW Processing
- Extended SHOWPLAN
- Very Extensive Testing

# Index Maintenance

## Minimize Page Splits with Fillfactor

- ☒ *Currently* very expensive as they "transactionally and exclusively" lock index pages.
- ☒ *Currently* very expensive as they flush all affected pages to disk immediately.
- ☑ Consider including re-establishing fillfactor on critical tables as part of normal operations

## v11 Optimizes Page Splits When Possible




- ☒ Pre-v11 split page in half causing 2 physical disk I/Os for all split pages. Major problem when loading data into clustered indexes!
- ☑ New algorithm splits page at insertion point. *If used properly*, eliminates unnecessary disk I/O and reduces pages splits.
- ☑ Makes loading data with clustered index on more viable in v11.
- ☑ Use dbcc tune(ascinsert, {1|0}, [table name] ) to turn on/off by table.
- ☑ Consider using this when:
  - ☑ Data is inserted in sequential order
  - ☑ If the table has clustered index on multiple columns and the "last" column increases monotonically.
- ☒ Do NOT ues this on data inserted in random order. Effect will only be increased disk space used with less rows per page.

# Optimization for DSS/Mixed Workload

- Optimize for Given Cache Configuration
  - Special Logic for Named Cache
  - fetch-and-discard Buffer Aging Strategy
- Extensive Subquery Enhancements
- Better Response-time Optimization
  - ORDER BY Queries
- Bigger Search Space for Large (>4) Joins
- Better Costing for Large Tables
- Isolation Level 0

# Tuning Transact-SQL Queries

## Enhanced SHOWPLAN in SQL Server 11


### ☑ ReWritten for Readability & v11 Feature Enhancements

- SELECT title_id FROM titles WHERE total_sales > ALL (SELECT total_sales FROM titles WHERE type = "business")

- QUERY PLAN FOR STATEMENT 1 (at line 1).
  STEP 1
  The type of query is SELECT
  FROM TABLE titles
  Nested iteration
  Table scan, Ascending scan, Positioned at start of table
  Run Subquery 1 (at nesting level 1)
  Using I/O Size 16kbytes, With LRU Buffer Replacement strategy
  NESTING LEVEL 1 SUBQUERIES FOR STATEMENT 1
  QUERY PLAN FOR SUBQUERY 1 (at nesting level 1 and at line 2)
  Correlated Subquery, Subquery under an ALL predicate
  STEP 1
  The type of query is SELECT
  Evaluate ungrouped ANY AGGREGATE
  FROM TABLE titles
  EXISTS TABLE: nested iteration
  Table scan, Ascending scan, Positioned at start of table
  Using I/O Size 16kbytes, With LRU Buffer Replacement strategy
  END OF QUERY PLAN FOR SUBQUERY 1

- Verify Proper Index Selections. Watch out for table scans.
- Verify Proper Join Order. Watch out for large tables.
- Verify Proper I/O Block Size *(only a recommendation)* and Cache Strategy.
- Verify Proper Subquery Optimizations
- Look at TEMPDB usage. May require tempdb optimizations for multi-user.



Copyright © Sybase, Inc., 1994-1995. All rights reserved.

# Dump Compatibility

- SQL Server 11.0 can Read 10.x Dumps
  - Not the dumps from 4.9x
- Upgrade During Db/Xn Load
  - Makes Selective Databases Upgrade Possible
  - *Eventually No Separate Upgrade Script*
- New Command to on-line/off-line Database
  - Limited Number of Commands when Off-line
- New Log Record Format with 11.0

# Install, Upgrade, and Configuration

- Upgrade from 4.9x or 10.x to 11.0
  - 10.x Dumps Upgraded During Db/Xn Load
- Improvements in Sybinit
  - Upgrade100 script gone
- New Configuration File
  - More Dynamic Configurable
  - Editable Config File
- Settable Object-level Attributes
  - New sysattributes system catalog

# System Configuration

- New sp_configure
  - Dynamic Reconfiguration
- Configure Parameter Hierarchy
  - Group Hierarchy
  - User Level Hierarchy, with sp_displaylevel
- Editable Config File
  - "Server won't boot" problems eliminated
  - buildmaster -y etc.. no longer necessary
  - Easy to Distribute Configuration

# Generalized Architecture Framework For Data Warehousing and Sybase Architecture Product Suite


**Ade Osinubi, PhD**
Principal Consultant
Sybase Worldwide Professional Services
Parsippany, New Jersey 07054
USA

## Agenda

- ▲ Definition of Data Warehousing / General Considerations
- ▲ Overview of SAFE Methodology
- ▲ Data Warehouse Business Problems and Drivers
- ▲ Generalized Architecture Requirements

## Agenda (contd)

- ▲ Generalized Data Warehouse Logical Architecture Framework
  - Real - Life Examples
- ▲ Generic Data Warehouse Product Suite
- ▲ Sybase Data Warehousing Products
  - Warehouse Works
  - Consulting and Architectural Services
- ▲ Conclusion

## What is a Data Warehouse?

"A Data Warehouse is an Architecture, Not a Product!"

*- The Gartner Group*

## Definition of A Data Warehouse

A Data Warehouse is a

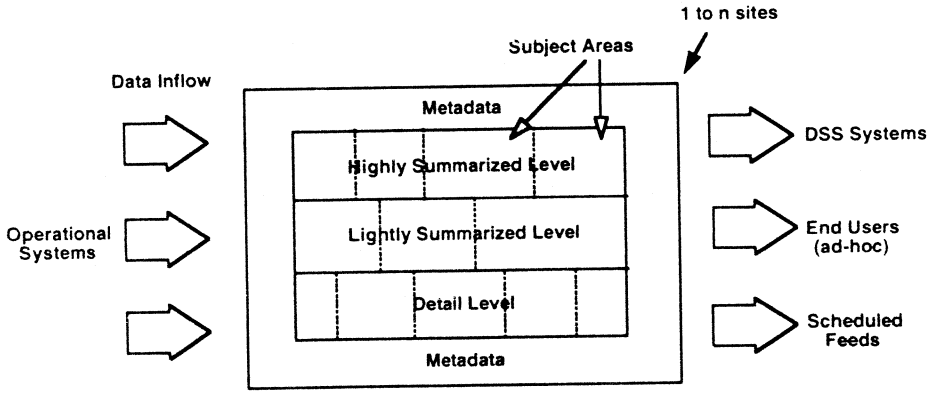
- subject oriented
- integrated
- non volatile
- time variant

collection of data in support of management's decision

*- W. H. Inmon*

## Conceptual Data Warehouse Model

-Ade Osinubi, 1995




Metadata are information about data

Copyright, Ade Osinubi

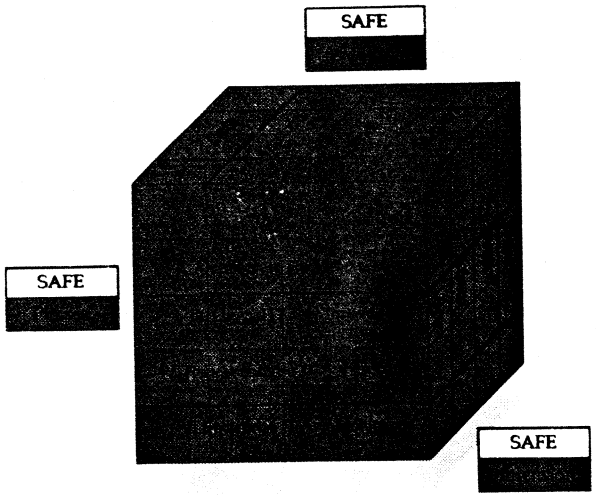
## Five Essential DW Components

- Operational Data Stores
- Data Conversion / Extraction
- Data Warehouse DBMS
- Data Warehouse Administration
- Business Intelligence Tools

## Key Design Considerations For Data Warehouse

- Granularity (most important aspect)
  - Dual Level of granularity applicable to most organizations
- Partitioning
  - breakup of data into independent units
- Data Warehouse follows CLDS life cycle
  - "give me what you have and I can tell you what I really want"
- Design for about 50% fit and then iterate or add increments

## Sybase Architecture Framework For The Enterprise (SAFE) Methodology




(c) 1995 Sybase, Inc. - Proprietary and Confidential

## Data Warehouse Business Problems and Drivers

- ▲ Reduce Cost
  - storage
  - charge back
- ▲ Eliminate Redundancy
- ▲ Improve Data Integrity
- ▲ Increase Revenue

## Data Warehouse Business Problems and Drivers (contd)

- Provide consistent and comprehensive access to customer information across multiple business areas
- To Better Understand Customers and meet their individual needs
- To meet Congressionally mandated requirements for reporting information and maintaining historical data

## Generic Architecture Requirements

- Data Architecture Requirements
- Application Architecture Requirements
- Technology Architecture Requirements
- Support Architecture Requirements

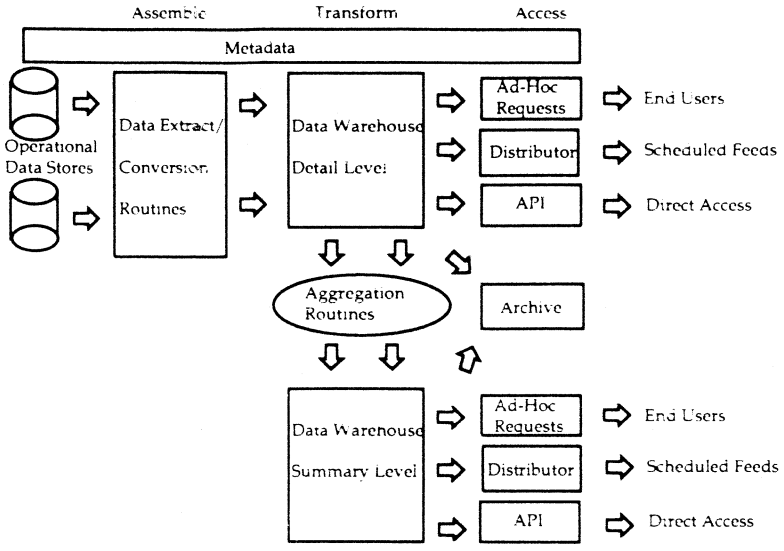
## Data Architecture Requirements

- Multiple Levels of Granularity of Data
- Very Large Data Volumes
  - could be terabytes in size
- Data Warehouse Levels Distributed or Centralized
- Synchronization of Multiple Data Sources

## Data Architecture Requirements (contd)

- Scheduled Feeds to Users
- Distribution of customized views to users
- Direct Retrievals From DW by Business Functions
- Extensible Data Model

## Application Architecture Requirements




## Technology Architecture Requirements

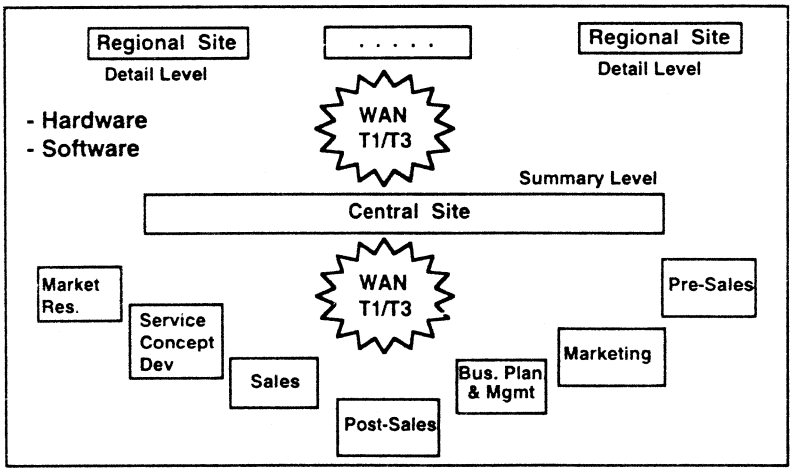
- ▲ Ability To Manage Large Data Volumes
- ▲ Ability To Manage Multiple Media
- ▲ Freely and Easily Index and Monitor Data
- ▲ Interfaces To Many Technologies

Gateway near under systems

## Technology Architecture Requirements (contd)

- Programmer/Designer Control of Data Placement
- Efficient Loading of Data and Data Access
  - Store and access data in parallel

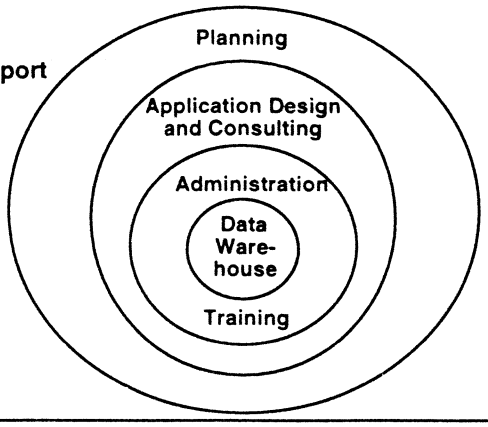
## Technology Architecture Requirements (contd)



## Support Architecture Requirememts

Multiple Functions:

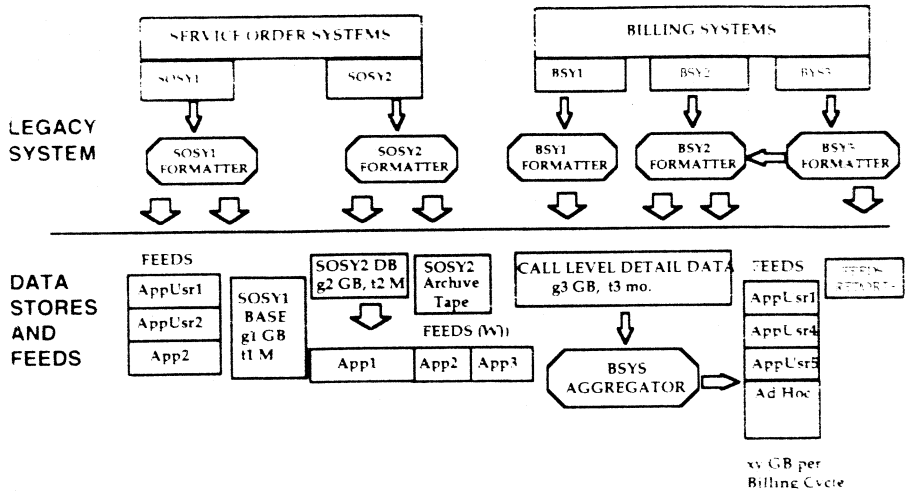
- Planning
- Design and Consulting Support
- Administration
- Training




## Generalized Data Warehouse Architecture Framework

- ▲ Logical Data Architecture
- ▲ Logical Application Architecture
- ▲ Logical Technology Architecture
- ▲ Logical Support Architecture

## Example Existing Environment




## Logical Data Architecture

- ▲ Enterprise Data Model
- ▲ Data Characteristics
- ▲ Data Storage
- ▲ Data Distribution

## Logical Data Architecture (contd)

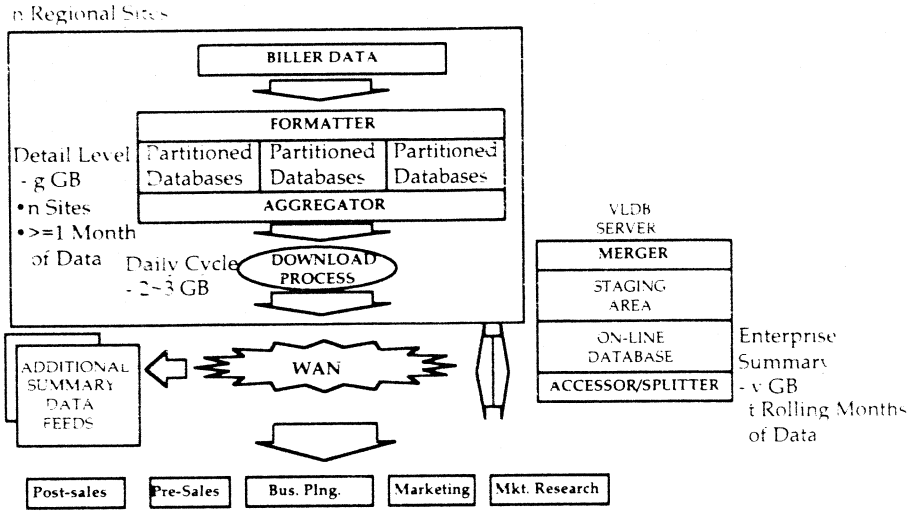
- ▲ Data Synchronization
- ▲ Data Integrity

Alleen zeer belangrijke tabellen Referentiële Integriteit

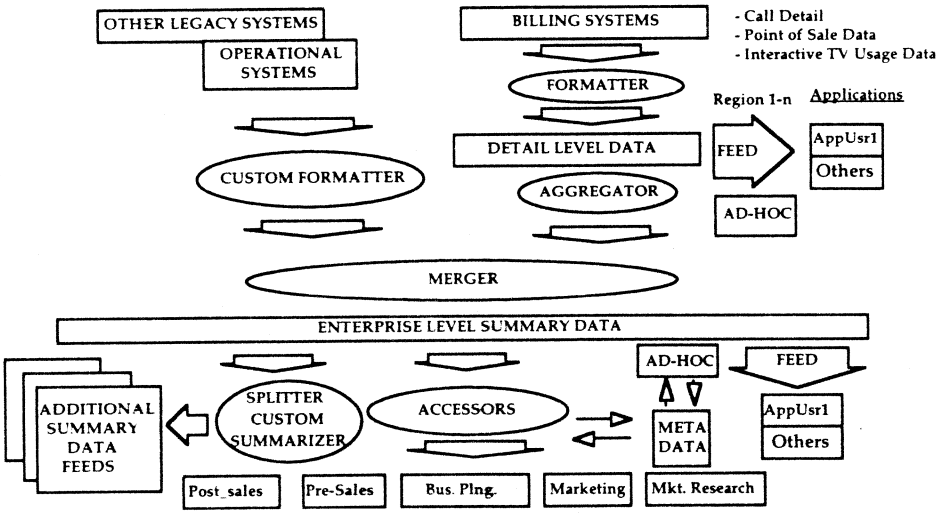
## Logical Data Architecture (contd): Enterprise Data Models

- ▲ Identify Subject Areas
- ▲ Data Model Approach
  - Relational
  - Multi-Dimensional or Star Schema

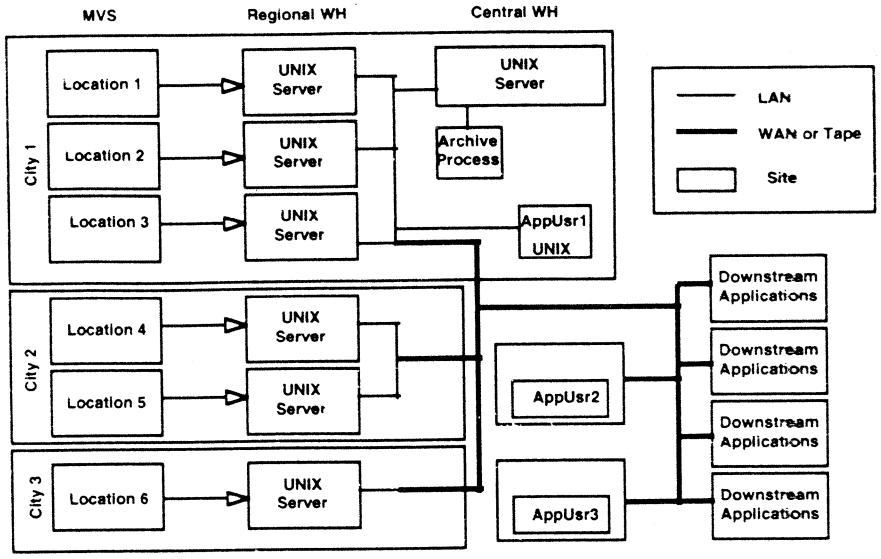
# Logical Data Architecture (contd)




# Logical Application Architecture



# Logical Technology Architecture



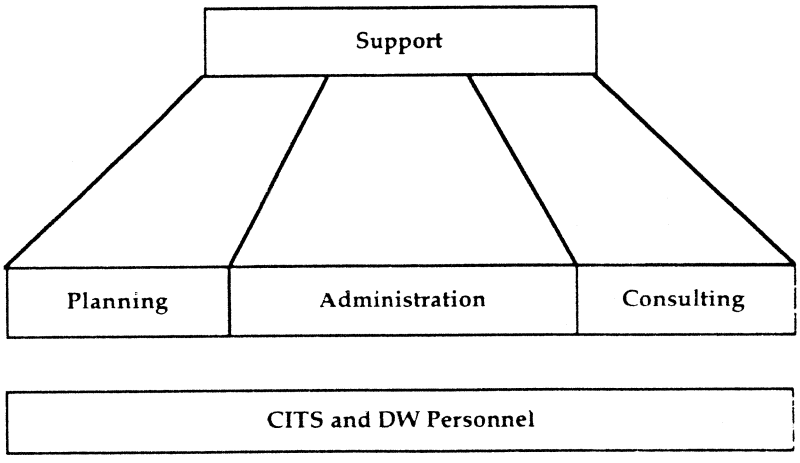

# Logical Support Architecture: Mission

- ▲ Facilitate Development
- ▲ Encourage Usage
- ▲ Sustain a production environment

## Logical Support Architecture: Delivery Strategy

- Combined Resources
- Seamless View to User Community
- Corporate ITS - day to day focus
- Data Warehouse Personnel - plan/consult

## Logical Support Architecture: Organization



## Logical Support Architecture: Deployment

- Headquarters
- Regional Sites
- CITS Sites

## Logical Support Architecture: Planning Function

- Capacity
- Integration
- Transformation Management
- Support Requirements Planning

## Logical Support Architecture: Planning Function (contd)

- Metadata Management
- External Communications

## Logical Support Architecture: Administration Function

- Systems Administration
- Warehouse Monitoring
- Capacity Monitoring
- Intermediate Data Store Management

## Logical Support Architecture: Administration Function (contd)

- DBMS Error Correction
- Metadata Maintenance
- Query Library Management
- Start Up Services

## Logical Support Architecture: Administration Function (contd)

- Problem Management
- Help Desk
- Security
- Change Control

## Logical Support Architecture: Administration Function (contd)

- Backup / Restore
- Software Distribution / Release Management
- Operations
- Performance Management

## Logical Support Architecture: Consulting Function

- User Assistance
- Data Model Development Coordination
- Query Creation Assistance
- Requirements

## Logical Support Architecture: Consulting Function (contd)

- Integration
- Planning and Education

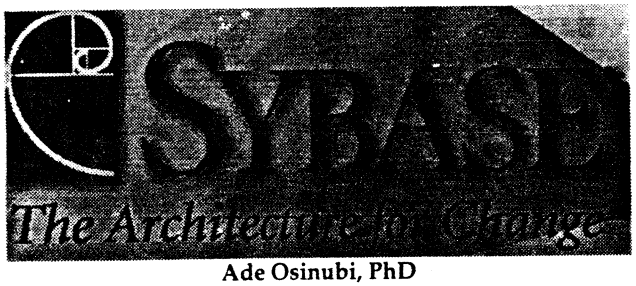
## Logical Support Architecture: Service Definitions

- Description
- Skills Required
- Training Required to meet skill requirement
- Procedures / Authorizations
- Materials / Forms

## Conclusion

- Data Warehouse Defined
- Data Warehouse Business Drivers Examined
- Generalized Data Warehouse Architecture Framework Developed Using SAFE
- Sybase Data Warehouse Products and Services Explored

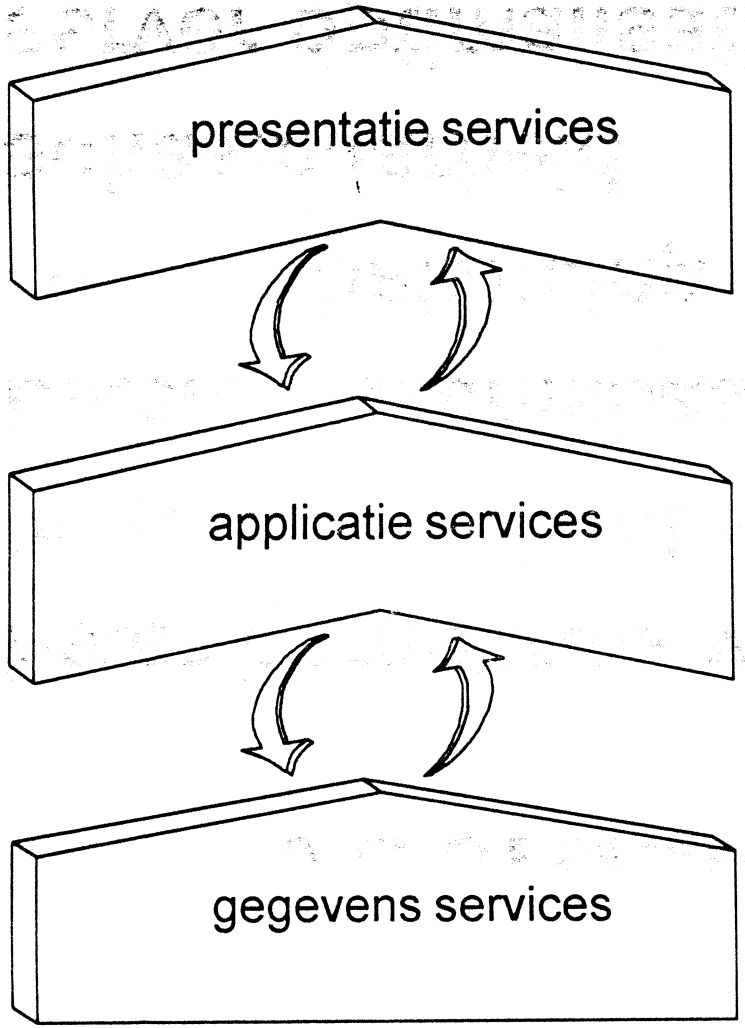
**Generalized Architecture Framework For Data Warehousing and Sybase Architecture Product Suite**

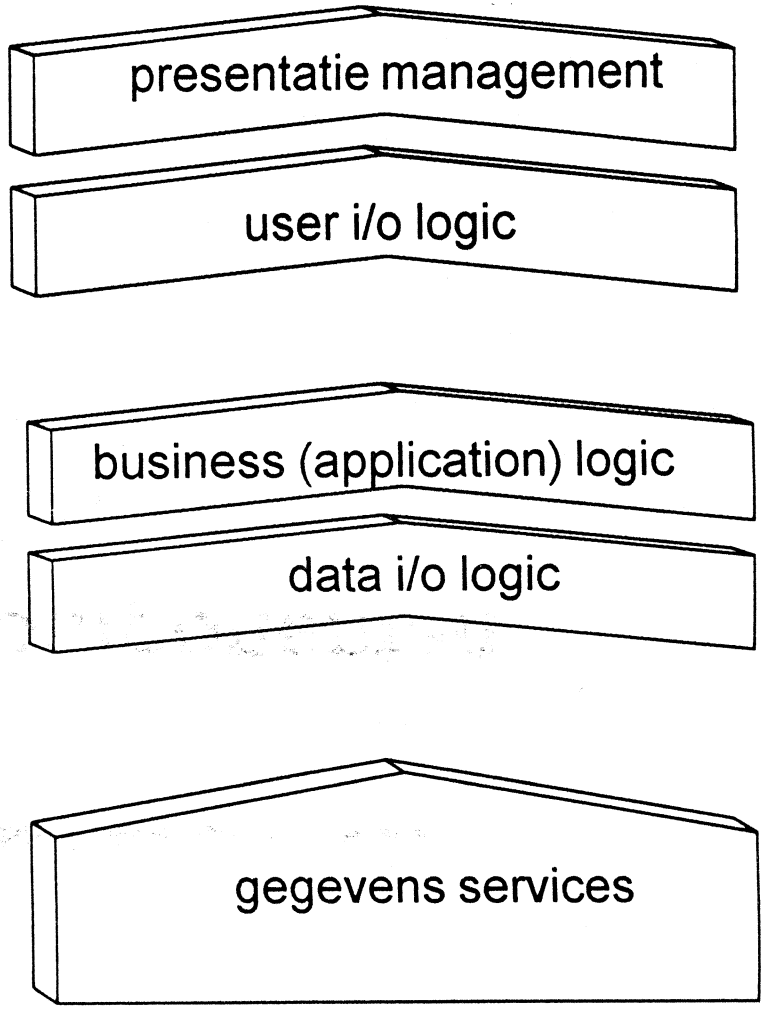



**Ade Osinubi, PhD**
Principal Consultant
Sybase Worldwide Professional Services
Parsippany, New Jersey 07054
USA

# 3-tier implementatie

# Software tiers



# Voordelen 3-tier ...

- beter hergebruik van componenten
- incrementeel ontwikkelen
- betere beschikbaarheid
  - load balancing
- schaalbaarheid
- server optimalisatie

# Sybase - DCE

- Security - kerberos
- Naming services
- Threads

# PowerBuilder 5 - CORBA

- Non visual objects via Object Request Broker
- Schaalbaarheid : Solaris, AIX, HP-UX
- Tool om objecten en methoden te selecteren aan de hand van tabellen en stored procedures

# Gevolgen Info Support

- Complexer
- Volgen ontwikkeling 3-tier gebied
- Specialisatie
  - Sybase (DCE)
  - PowerBuilder
- TI Composer, HPS, OEC Entera ...