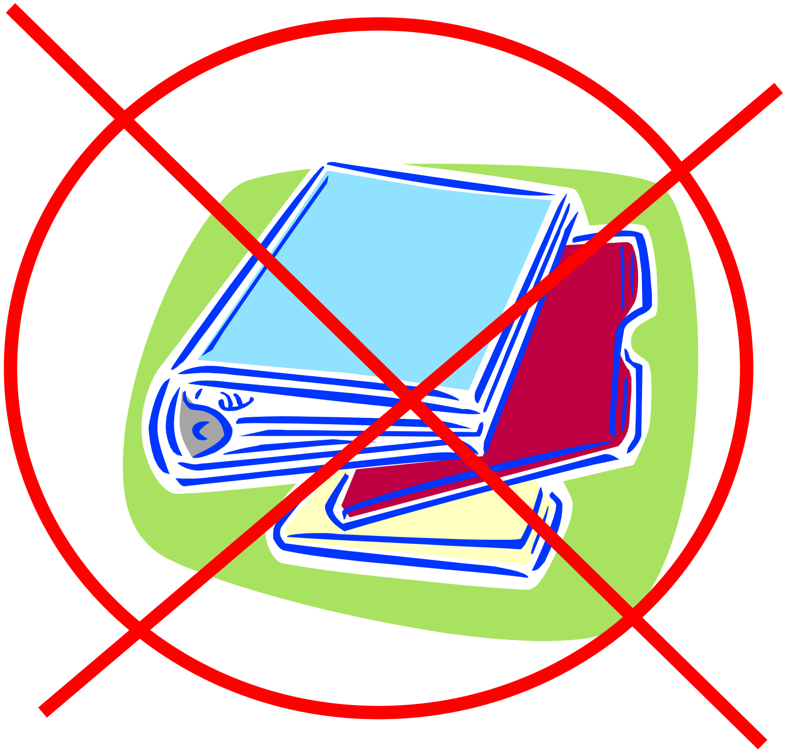
Improving IDS Administration Zone

John F. Miller III
IBM Senior Technical Staff Member

Administration Free Zone

- **Low Total Cost of Ownership and Deployment**
- **Easy Scalable Administration**
- **Open Admin API for customized administration**
- **Autonomic architecture**
- **Industrial strength, highly reliable - Install it, Set it up, and Forget about it.**



Create Index Automatically Creates Distributions & Statistics

Enhanced dbinfo()

Global configuration of non-logging temp tables

Multiple Trigger per table

sysdbopen & sysdbclose

Automatic Checkpoints

Private Memory Caches for Virtual Processors

Auto AIO VP Tuning

OnBar Whole System Parallel Restore

Automatic real-time statistics created for temp tables

Auto LRU Tuning

Database Restart Policy

Enhanced Explain

Non-Block Checkpoints

SQL History Tracing

Database Scheduler

SQL Admin API

ontape Directory Backup

Online Reconfiguration of the Physical log

Last Committed Isolation

Dynamically Rename Objects in ER

ANSI Join Directives

Enhanced sysmaster

Encrypted Communication For HDR

Detailed Thread Wait Information

Overview

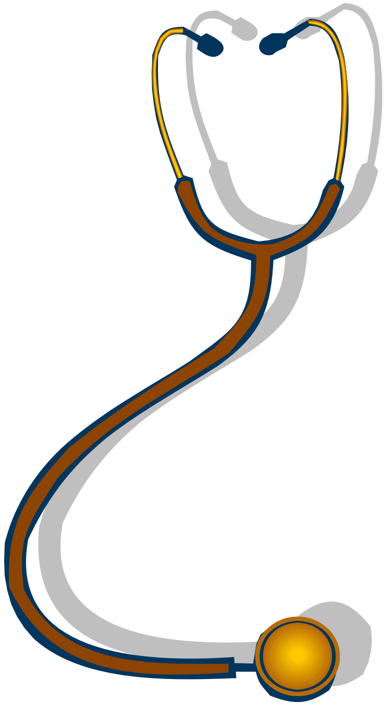
- **SQL Based Administration Commands**
- **Improved Sysmaster Database**
- **SQL History Tracing**
- **Built in Database Scheduler**
- **Quick overview of IDSAdmin**

Laying the Foundation for Improvements

- **A new database (sysadmin) for**
  - Administrative functions
  - Alert System
  - Data collection
- **Different methods for collecting information**
- **Additional information required to present in a way DBAs can understand**

Improve Information for sysmaster and onstat

- **Improve the display of information in the following areas:**
  - Improved Thread Wait Informix
  - Checkpoint Information
  - User/Server Environment Informix
  - Online & onbar log files
  - Network Information (IO counts and times)
  - MGM Information
  - SQL Generic Cache Profiles
  - SQL Statement History



Make the information clear for the DBA to understand

New Sysmaster Tables

TABLE NAME	DESCRIPTION
syscheckpoint	**The information about the checkpoint and associated statistics**
systcblst	**Modified the existing table to add wait stats.**
sysenvses	**View Informix's session environment variables**
sysenv	**View the servers environment variables**
sysonlinelog	**View the online.log for the server**
sysscblst	**Improvement to view the memory used by session**
sysnetworkio	**View the network I/O generated by database session**
sysdual	**Oracle compatibility feature**
syssqlcacheprof	**Displays the profile information about each SQL cache**
syssqltrace	**The sql statements which have been recently executed on the system**
syssqltrace_itr	**The list of iterators for the SQL statement.**
syssqltrace_info	**General information about the SQL tracing**
sysnetglobal	**Global Network Information**
sysnetclienttype	**Network information based on client type**
sysbaract_log	**The OnBar Activity Log file**
sysrstcb	**Improvement to view I/O and lock wait information**

Onstat –g ath

- **Remove/Reduce the number of thread status of “sleeping forever”**
- **Give the DBA a clear picture of what is happening**

```
Threads:
tid    tcb            rstcb        prty status               vp-class   name
2      10bbf36a8       0             2    sleeping forever      3lio    lio vp 0
3      10bc12218       0             2    sleeping forever      4pio    pio vp 0
4      10bc31218       0             2    sleeping forever      5aio    aio vp 0
5      10bc50218       0             2    sleeping forever      6msc    msc vp 0
6      10bc7f218       0             2    sleeping forever      7aio    aio vp 1
7      10bc9e540       10b231028     4    sleeping secs: 1      1cpu    main_loop()
8      10bc12548       0             2    running               1cpu    tlitcppoll
9      10bc317f0       0             3    sleeping forever      1cpu    tlitcplst
10     10bc50438       10b231780     2    sleeping forever      1cpu    flush_sub(0)
11     10bc7f740       0             2    sleeping forever      8aio    aio vp 2
12     10bc7fa00       0             2    sleeping forever      9aio    aio vp 3
13     10bd56218       0             2    sleeping forever     10aio    aio vp 4
14     10bd75218       0             2    sleeping forever     11aio    aio vp 5
15     10bd94548       10b231ed8     3    sleeping forever      1cpu    aslogflush
16     10bc7fd00       10b232630     1    sleeping secs: 26     1cpu    btscanner 0
32     10c738ad8       10b233c38     4    sleeping secs: 1      1cpu    onmode_mon
50     10c0db710       10b232d88     2    cond wait  netnorm    1cpu    sqlexec
```

Improved onstat –g ath

```
Threads:
 tid     tcb             rstcb        prty status                vp-class      name
  2       10bbf36a8       0             2     IO Idle                3lio        lio vp 0
  3       10bc12218       0             2     IO Idle                4pio        pio vp 0
  4       10bc31218       0             2     running                5aio        aio vp 0
  5       10bc50218       0             2     IO Idle                6msc        msc vp 0
  6       10bc7f218       0             2     running                7aio        aio vp 1
  7       10bc9e540       10b231028     4     sleeping secs: 1       1cpu        main_loop()
  8       10bc12548       0             2     running                1cpu        tlitcppoll
  9       10bc317f0       0             3     sleeping forever       1cpu        tlitcplst
  10      10bc50438       10b231780     2     IO Wait                1cpu        flush_sub(0)
  11      10bc7f740       0             2     IO Idle                8aio        aio vp 2
  12      10bc7fa00       0             2     IO Idle                9aio        aio vp 3
  13      10bd56218       0             2     IO Idle               10aio        aio vp 4
  14      10bd75218       0             2     IO Idle               11aio        aio vp 5
  15      10bd94548       10b231ed8     3     sleeping forever       1cpu        aslogflush
  16      10bc7fd00       10b232630     1     sleeping secs: 34      1cpu        btscanner 0
  32      10c738ad8       10b233c38     4     sleeping secs: 1       1cpu        onmode_mon
  50      10c0db710       10b232d88     2     IO Wait                1cpu        sqlexec
```

SQL Admin Commands

- **A set of User Defined Routines (UDRs) to administer the Informix database server.**
- **The major categories of administration include:**
  - Space Management
  - Configuration Management
  - Routine task maintenance
  - System Validation (oncheck functionality)
- **Feature Benefits**
  - SQL Based Administration
  - Remote Administration
  - Tracking of command execution and results in a system table

Admin Commands – Two New UDRs

- **Two UDRs called *task* & *admin* are part of the sysadmin database**
- **They perform exactly the same, only the return code is different**
  - *task()* UDR returns a character string describing the return status

```
EXECUTE FUNCTION task('create dbspace', 'dbspace2', '/CHUNKS/dbspace2');
(expression)  created dbspace number 2 named dbspace2
```

  - admin() UDR returns a integer return status which is a link to the *command_history* table

```
EXECUTE FUNCTION admin('create dbspace', 'dbspace2', '/CHUNKS/dbspace2');
(expression)  107
```

- **Both UDRS log all executions into a table called *command_history in the sysadmin database***

SQL Admin Command - Tracking

- **SQL Admin API logs all executions into a table called *command_history in the sysadmin database***

Column	Type	Description
cmd_number	serial	
cmd_exec_time	datetime year to second	Time the command was started
cmd_user	varchar	User executing the command
cmd_hostname	varchar	Host the command was executed from
cmd_executed	varchar	The command executed
cmd_ret_status	Integer	Return code
cmd_ret_msg	lvarchar	Return message

SQL Admin Commands - Parameters

- **Environment Variable Expansion**
  - A pathname may start with an environment variable.
  - The environment variable may only exist in the server’s environment

- **Unit Extensions**
  - All offsets and sizes can be provided with unit extensions
  - The extensions are case insensitive
  - Default is KB
  - PB, TB, GB, MB, KB, B

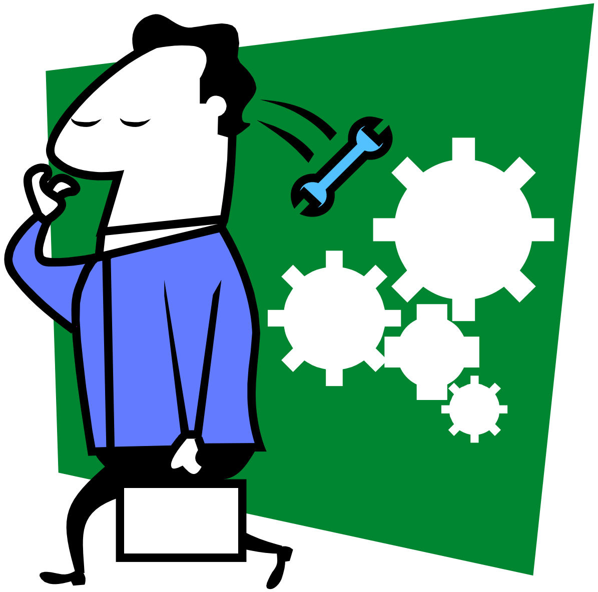
SQL Admin Commands – EXAMPLES

```
EXECUTE FUNCTION admin('create dbspace','dbspace2',‘$INFORMIXDIR/SPACE/dbspace2',`20MB`);
(expression)         108

SELECT * FROM command_history WHERE cmd_number IN (108)
cmd_number      108
cmd_exec_time   2005-11-17 16:26:15
cmd_user        informix
cmd_hostname    olympia.beaverton.ibm.com
cmd_executed    create dbspace
cmd_ret_status  0
cmd_ret_msg     created dbspace number 2 named dbspace2
```

DBA Constant Struggle

Identifying Performance Bottlenecks in SQL Statements

- **Current methods for examination**
  - Set explain
  - Looking at database objects accessed by the SQL for inefficiency
- **Lack of simplicity in the process**
- **Hard to build a repeatable process**



Questions DBAs like to ask?

- **How long did a SQL statements take?**
- **How many resources of each category did a statement take?**
  - Disk I/O
  - Memory
  - CPU
- **How long and how many times did we wait on each resource?**
  - Locks
  - Disk I/O

Solutions – SQL Query Drill Down Feature

- **Provide consolidated detail information about SQL statements through all layers**
- **Information available through onstat or sysmaster database**
- **Dynamically configurable**
- **By default disabled**
- **Global and User Tracing modes**

Name	Description
User ID	**User id**
Session ID	**Database Session ID**
Database Name	**Current Database Name**
Statement Type	**Type of SQL statement being executed**
Statement Execution time	**Duration of the SQL statement**
Time of Execution	**Date & Time this statement completed**
Statement Text	**SQL statement text or the procedure stack trace with statement type**
RSAM statistics	• **Buffer Reads & Writes** • **Page reads & Writes** • **Memory Sorts, disk Sorts,** • **Lock requests, waits** • **Logical Log Records** • **Index buffer reads**
SQL statistics	• **Estimated # of rows** • **Estimated Cost** • **# of rows returned** • **Statement Type** • **Database Isolation Level**

Controlling SQL Query Drill Down

- **ONCONFIG variable SQLTRACE**
  - Level =[off,low,med,high]
  - Ntraces=[number of traces]
  - Size=[size of each trace buffer in KB]
  - Mode=[global|user]

```
SQLTRACE level=low,ntraces=2000,size=1,mode=global
```

- **Turn off SQL Tracing for session id 147**

```
execute function task("SET SQL USER TRACING OFF",147);
```

Controlling SQL Query Drill Down

- **Dynamically enable or modify SQL Tracing**
  - Trace 2000 SQL statements
  - Trace 1024 bytes of data for each SQL statement

```
execute function task("SET SQL TRACING ON",2000,1);
```

- **Turn off SQL Tracing**

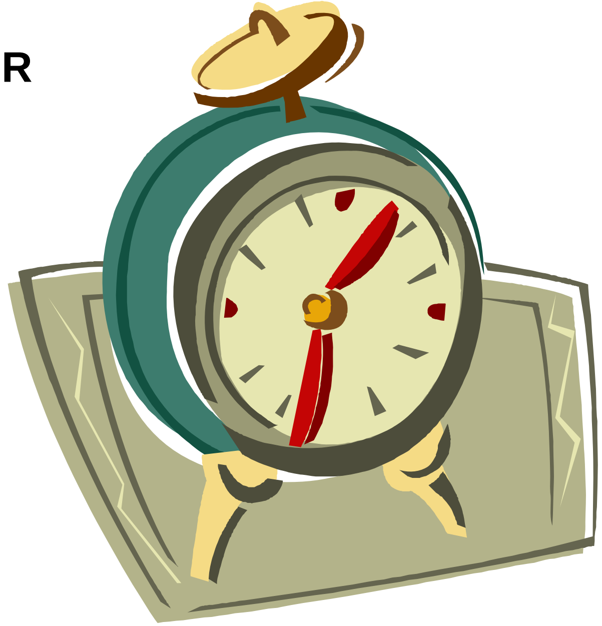
```
execute function task("SET SQL TRACING OFF");
```

SQL Query Drill Down - onstat

```
Database:        sysadmin
 Statement text:
  SELECT MAX(run_task_seq) FROM ph_run A, ph_task B WHERE A.run_task_id = ?
    AND A.run_task_id = B.tk_id AND A.run_time + B.tk_delete < CURRENT

 Iterator/Explain
 ================
    ID   Left  Right   Est Cost   Est Rows   Num Rows     Type
     3      0      0          1          1          1     Index Scan
     4      0      0         19        545          1     Index Scan
     2      3      4         20          5          1     Nested Join
     1      2      0          1          1          1     Group

 Statement information:
  Sess_id  User_id  Stmt Type          Finish Time    Run Time
  21       0        SELECT             10:51:11       0.0023

 Statement Statistics:
  Page       Buffer     Read       Buffer     Page       Buffer     Write
  Read       Read       % Cache    IDX Read   Write      Write      % Cache
  0          77         100.00     0          0          0          0.00

  Lock       Lock       LK Wait    Log        Num        Disk       Memory
  Requests   Waits      Time (S)   Space      Sorts      Sorts      Sorts
  0          0          0.0000     0.000 B    0          0          0

  Total      Total      Avg        Max        Avg        I/O Wait   Avg Rows
  Executions Time (S)   Time (S)   Time (S)   IO Wait    Time (S)   Per Sec
  220        78.8463    0.3584     1.9557     0.000000   0.000000   439.9908

  Estimated  Estimated  Actual     SQL        ISAM       Isolation  SQL
  Cost       Rows       Rows       Error      Error      Level      Memory
  20         1          1          0          0          DR         41552
```

Built-in Database Scheduler

- **Ability to schedule SQL, Stored procedures or a UDR**
- **The schedule entities are called “tasks”**
- **There are different types of tasks**
  - Tasks
  - Sensor
  - Startup Task
  - Startup Sensor
- **Tasks are driven by the data inside a table called** *ph_task*



- **Sensors are a specialized task designed to collection information**
  - Easy to add and configure
  - Collect information and stores it in database tables

Scheduler Specialized Task

- **Task**
  - A means to execute a specific job at a specific time or interval
  - The *task* executes by invoking
    - A single or compound SQL statement
    - Stored procedure,
    - C User Defined Routine
    - Java User Defined Routine

Sensor - Specialized Task

- **A specialized task geared at collecting and saving data without the DBA doing much work.**
- **A means to get information about a managed element**

```
INSERT INTO ph_task  (tk_name,tk_type,tk_group,tk_description,tk_result_table,tk_create,
                      tk_execute,tk_start_time,tk_stop_time,tk_frequency,tk_delete)
VALUES
("mon_profile",
"SENSOR",
"PERFORMANCE",
"Collect the general profile information",
"mon_prof",
"create table mon_prof (ID integer, number integer, value int8 );",
"insert into mon_prof select $DATA_SEQ_ID, number, value from sysmaster:sysshmhdr where name != 'unused'",
DATETIME(06:00:00) HOUR TO SECOND,
DATETIME(18:00:00) HOUR TO SECOND,
INTERVAL ( 30 ) MINUTE TO MINUTE,
INTERVAL ( 30 ) DAY TO DAY);
```

Why use Tasks and Sensors?

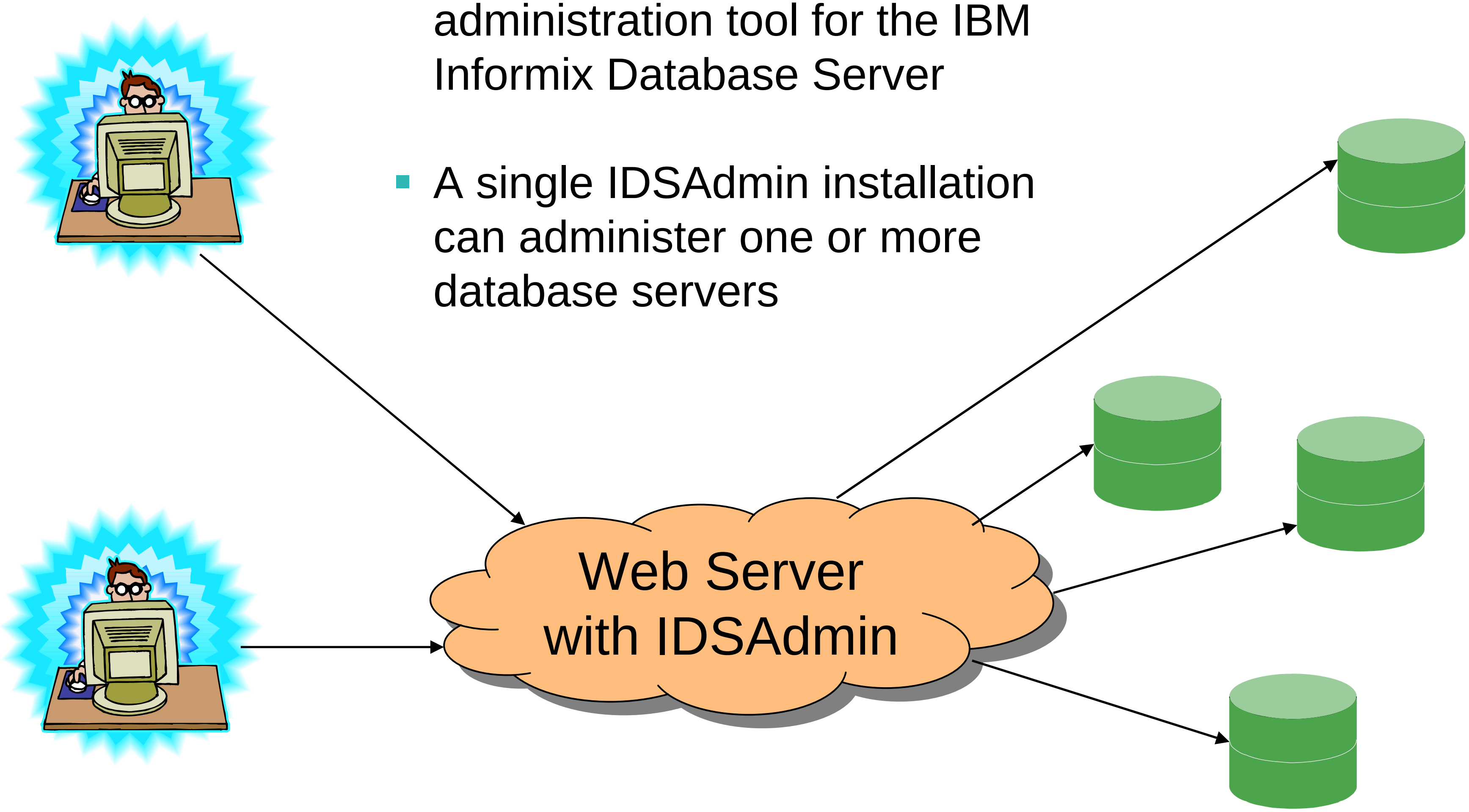
- **Build a specialized task**
  - A way of ensuring routine jobs get completed
  - Periodically check and/or analyze collected data to ensure the data server is operating efficiently

- **Sensors provide a simple way of collecting information**
  - Easy to add a new sensor
  - Provide a portable way of collecting information without using the operating system.

Admin Console

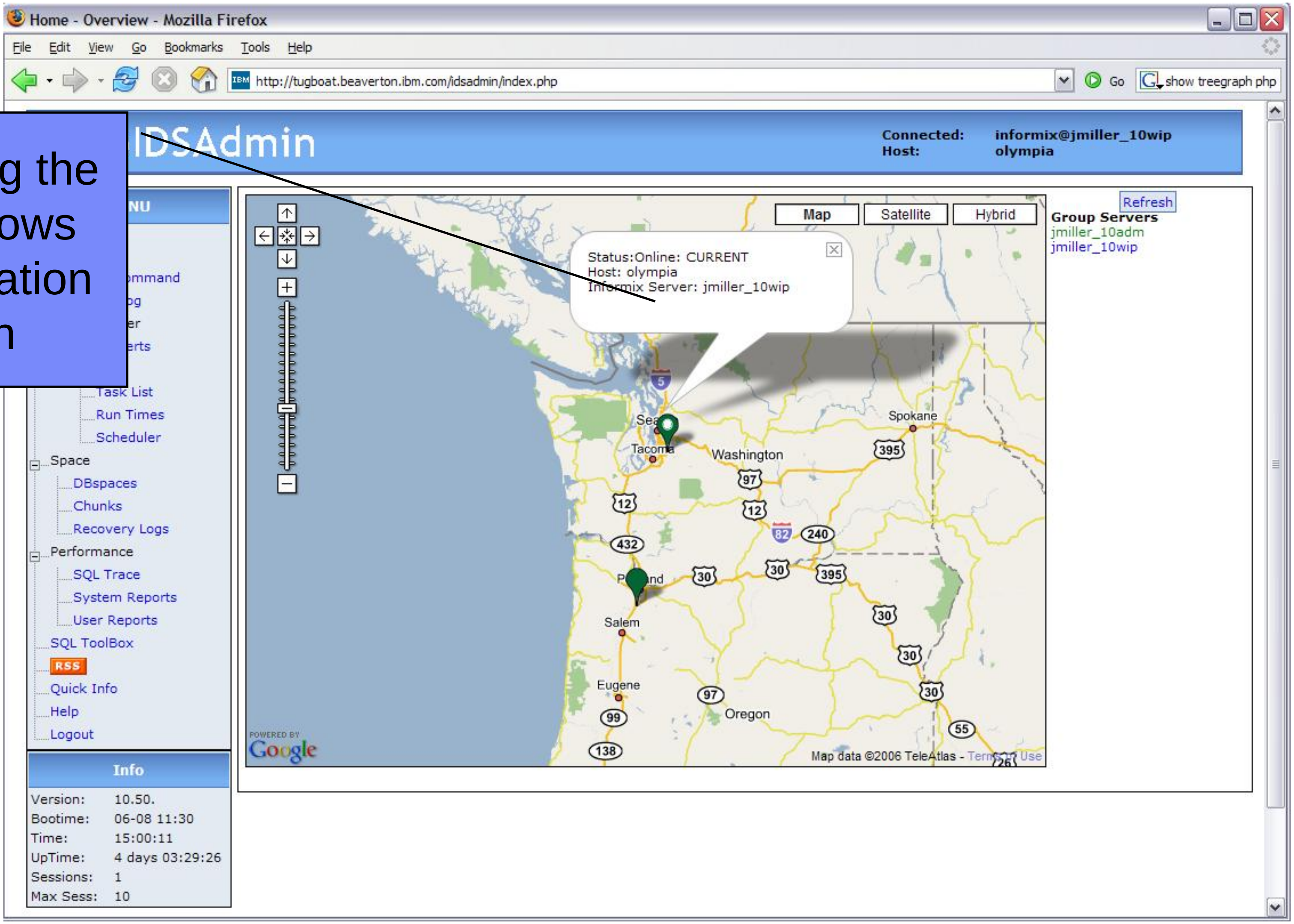
What is IDSAdmin?

- IDSAdmin is a web-based administration tool for the IBM Informix Database Server
- A single IDSAdmin installation can administer one or more database servers



Map View

Highlighting the server shows the information balloon



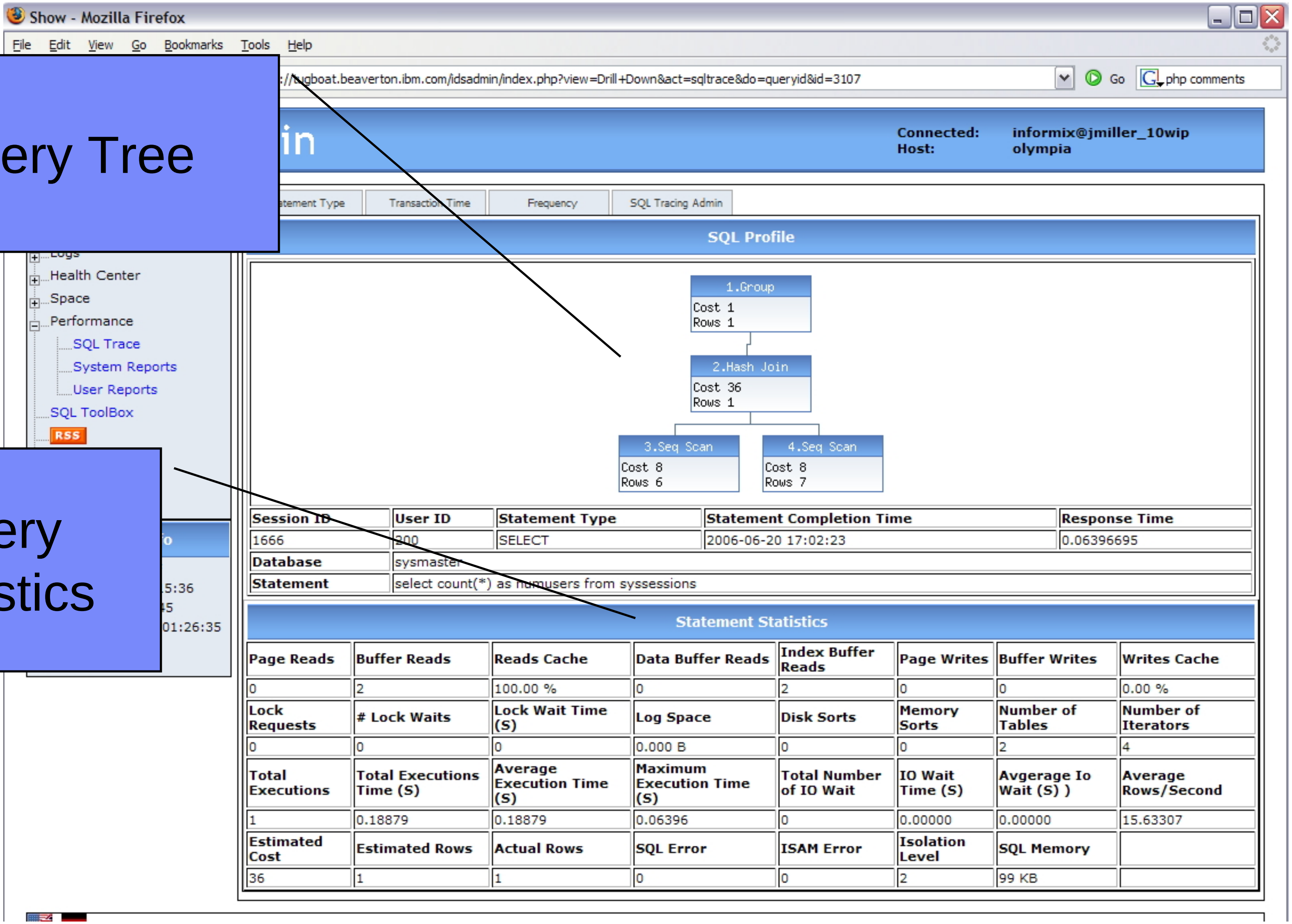
Connected: informix@jmiller_10wip
Host: olympia

Status:Online: CURRENT
Host: olympia
Informix Server: jmiller_10wip

Info

Version: 10.50.
Bootime: 06-08 11:30
Time: 15:00:11
UpTime: 4 days 03:29:26
Sessions: 1
Max Sess: 10

A Specific Instance of an SQL Statement



SQL Profile

1.Group
Cost 1
Rows 1

2.Hash Join
Cost 36
Rows 1

3.Seq Scan
Cost 8
Rows 6

4.Seq Scan
Cost 8
Rows 7

Session ID	User ID	Statement Type	Statement Completion Time	Response Time
1666	200	SELECT	2006-06-20 17:02:23	0.06396695
Database	sysmaster			
Statement	select count(*) as numusers from syssessions			

Statement Statistics

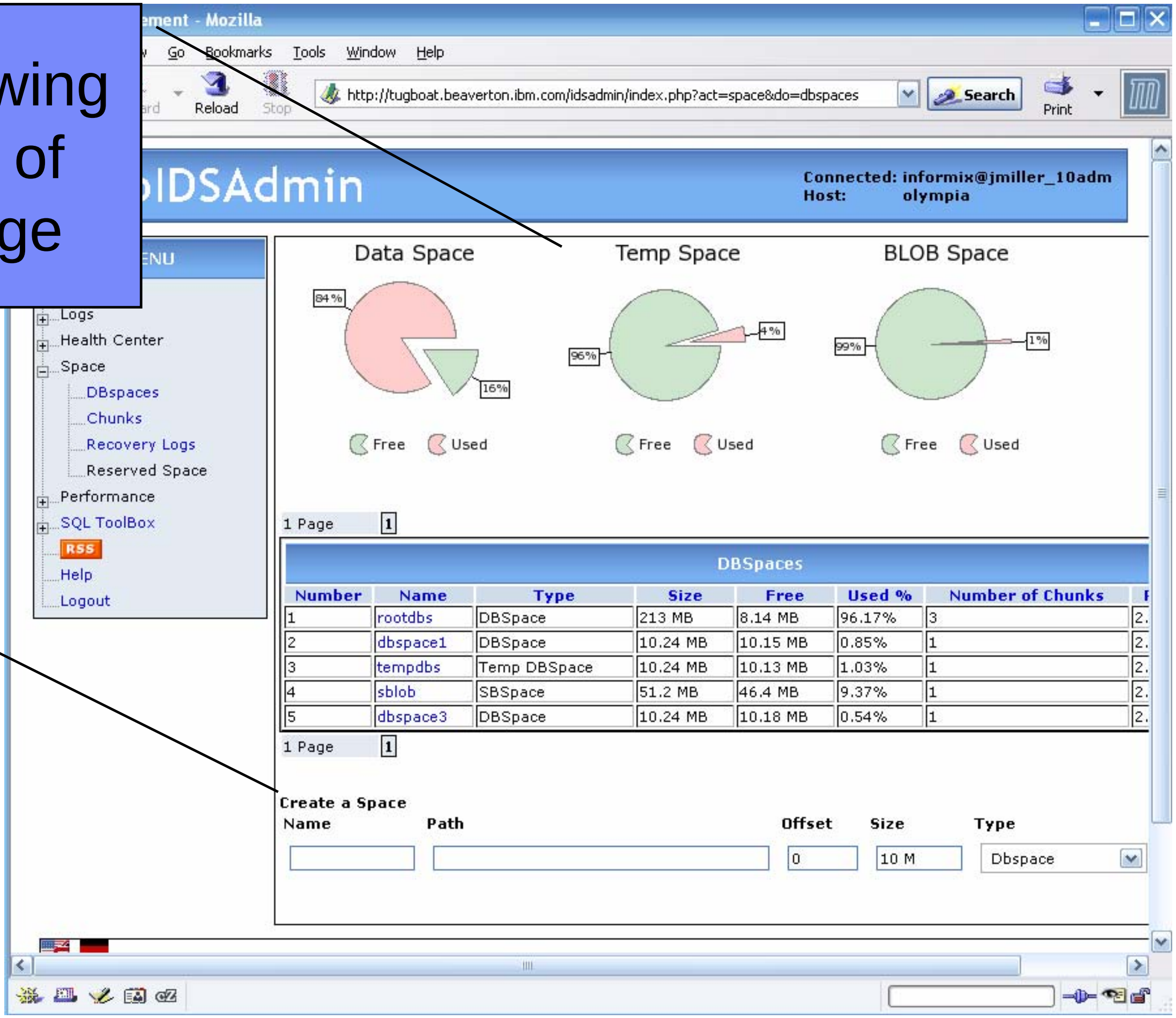
Page Reads	Buffer Reads	Reads Cache	Data Buffer Reads	Index Buffer Reads	Page Writes	Buffer Writes	Writes Cache
0	2	100.00 %	0	2	0	0	0.00 %
Lock Requests	**# Lock Waits**	**Lock Wait Time (S)**	**Log Space**	**Disk Sorts**	**Memory Sorts**	**Number of Tables**	**Number of Iterators**
0	0	0	0.000 B	0	0	2	4
Total Executions	**Total Executions Time (S)**	**Average Execution Time (S)**	**Maximum Execution Time (S)**	**Total Number of IO Wait**	**IO Wait Time (S)**	**Avgerage Io Wait (S))**	**Average Rows/Second**
1	0.18879	0.18879	0.06396	0	0.00000	0.00000	15.63307
Estimated Cost	**Estimated Rows**	**Actual Rows**	**SQL Error**	**ISAM Error**	**Isolation Level**	**SQL Memory**	
36	1	1	0	0	2	99 KB	

Query Tree

Query Statistics

Overview of Disk Space

Graphs showing quick view of space usage



http://tugboat.beaverton.ibm.com/idsadmin/index.php?act=space&do=dbspaces

Connected: informix@jmiller_10adm
Host: olympia

DBSpaces

Number	Name	Type	Size	Free	Used %	Number of Chunks	F
1	rootdbs	DBSpace	213 MB	8.14 MB	96.17%	3	2.
2	dbspace1	DBSpace	10.24 MB	10.15 MB	0.85%	1	2.
3	tempdbs	Temp DBSpace	10.24 MB	10.13 MB	1.03%	1	2.
4	sblob	SBSpace	51.2 MB	46.4 MB	9.37%	1	2.
5	dbspace3	DBSpace	10.24 MB	10.18 MB	0.54%	1	2.

1 Page 1

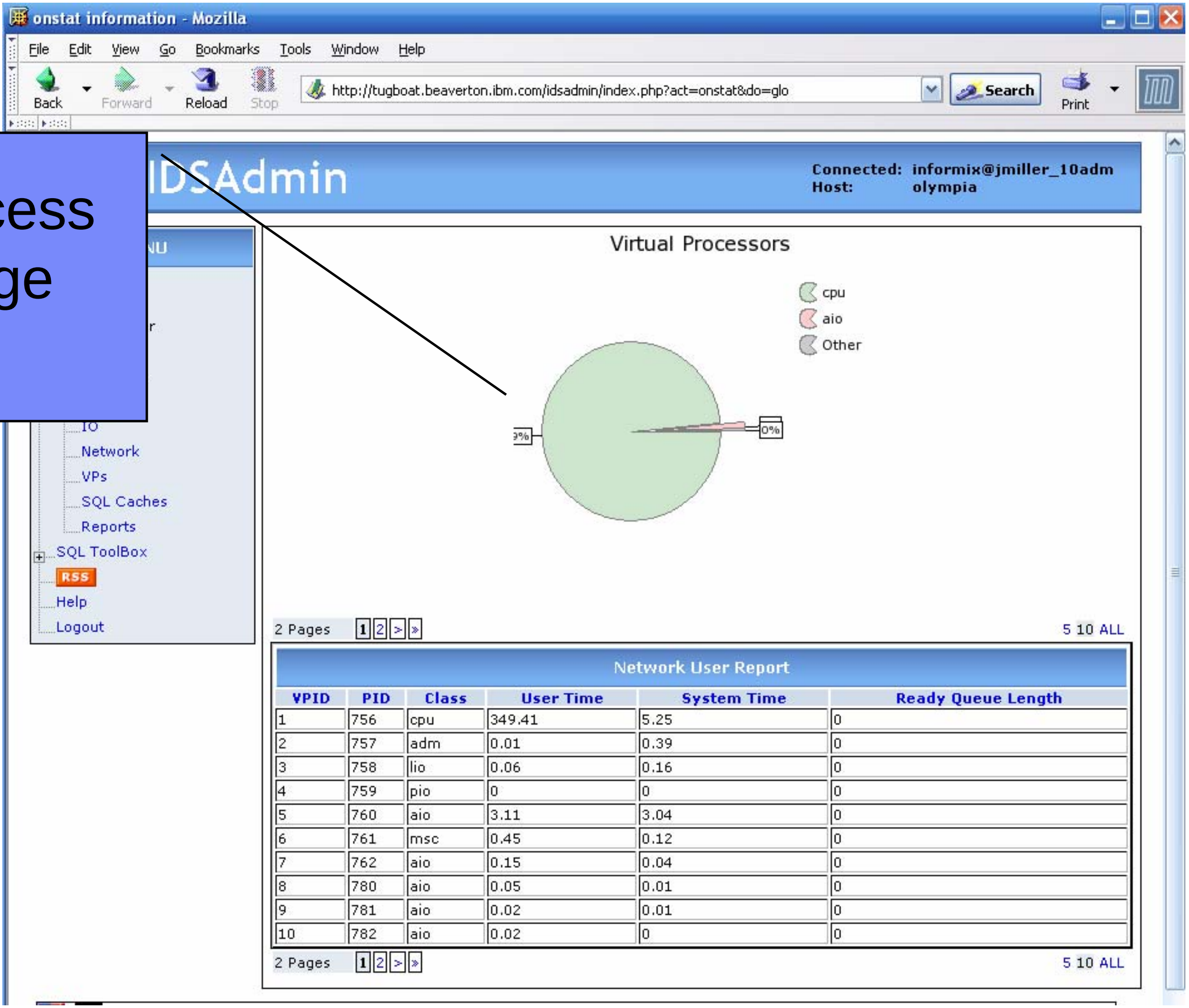
Create a Space

Name	Path	Offset	Size	Type
		0	10 M	Dbspace

Ability to create additional spaces

Process Usage

Virtual Process CPU Usage Graph



Network User Report

VPID	PID	Class	User Time	System Time	Ready Queue Length
1	756	cpu	349.41	5.25	0
2	757	adm	0.01	0.39	0
3	758	lio	0.06	0.16	0
4	759	pio	0	0	0
5	760	aio	3.11	3.04	0
6	761	msc	0.45	0.12	0
7	762	aio	0.15	0.04	0
8	780	aio	0.05	0.01	0
9	781	aio	0.02	0.01	0
10	782	aio	0.02	0	0

Create Index Automatically Creates Distributions & Statistics

Enhanced dbinfo()

Global configuration of non-logging temp tables

Multiple Trigger per table

sysdbopen & sysdbclose

Automatic Checkpoints

Private Memory Caches for Virtual Processors

Auto AIO VP Tuning

OnBar Whole System Parallel Restore

Automatic real-time statistics created for temp tables

Auto LRU Tuning

Database Restart Policy

Enhanced Explain

Non-Block Checkpoints

SQL History Tracing

Database Scheduler

SQL Admin API

ontape Directory Backup

Online Reconfiguration of the Physical log

Last Committed Isolation

Dynamically Rename Objects in ER

ANSI Join Directives

Enhanced sysmaster

Encrypted Communication For HDR

Detailed Thread Wait Information