BASIC SERVER PERFORMANCE TUNING

RAID 10!

DB SERVER HARDWARE

- Some best practice considerations for hardware:
  - 64 bit cpu! More memory, more registers!
  - Tons of RAM!
  - 4-8 cpu cores max - more is not always better because of concurrency contention issues, though as MySQL improves, this guideline might change.
  - RAID everywhere.
    - RAID 10 for MySQL data directory - best performance.
    - RAID 1 or 5 for the operating system

64 BIT, EH?

- MySQL is threaded - PAE (Physical Address Extension) doesn't work very well:
  - Each thread in MySQL could only use about 2.5GB of RAM
  - That means global buffers could not be larger than 2.5GB at best.
  - Performance would suffer because of the additional overhead from PAE
- 8 more GPR/SSE registers in 64-bit

RAID COMPARISON

	Min # of Drives	Capacity	Fault Tolerance	Random Reads	Random Writes	Seq. Reads	Seq. Writes
RAID0	1	N	NONE	Good	Fast	Fast	Fast
RAID1	2	1	N-1	Fast	Good	Fast	Good
RAID5	3	N-1	1	OK	Bad	Good	OK
RAID10	4	N/2	1+	Fast	Fast	Fast	Fast

OPERATING SYSTEM

- As mentioned previously, the operating system should preferably be installed on some sort of RAID storage, and in a perfect world separate from the database RAID system.
- Also preferred to use Linux, but if you have to use Windows, so be it. ;)
- As for flavor/version, there isn't a terrible amount of concern here, so long as you have a relatively recent 64 bit kernel and any additional features you need.

ADDITIONAL OS CONSIDERATIONS

- Care should be taken to ensure the database is properly started and stopped with the operating system state.
- Logs should be monitored and rotated as necessary.
- Simple system and database monitoring scripts can be utilized to alert on issues including:
  - Server and database health
  - Low disk space
  - Poor performance

MYSQL SERVER TUNABLES

- There are many, many variables in the MySQL server available for tuning and tweaking.
- With minor adjustments on a handful of these variables, one can often optimize the server to within a few percent of perfect. Beyond this point, additional changes will require extensive benchmarking and analysis to squeeze a minimal amount of additional performance.
- We will cover the major options here; consult a google beyond that. :)
- After this point, database structure and query structure should be scrutinized, which is the topic of the following lecture.

SEEING DEFAULT SETTINGS

- To see all of the default settings for the tuning variables:
  - `/usr/libexec/mysqld --verbose --help`
- This will produce a list of every tunable variable parameter that you can plug in to `my.cnf`, as well as a long table of default values for every setting.
- An **excellent** resource documenting all parameters, including those not available via the command line, is section 5.1.3.

SEEING CURRENT SETTINGS

- To see all of the current settings for the tuning variables:
  - `SHOW` *`[GLOBAL]`* `VARIABLES;`
- This will show the *current* running values for all of the tuning variables for the current session. The `GLOBAL` parameter shows server-wide settings.
- Note that some of these settings can be changed dynamically with:
  - `SET` *`[GLOBAL]`* `name = value;`

FIRST, CHECKING STATUS

- Before tuning values, it is important to consider the server's current *status* and operation metrics.
  - `SHOW GLOBAL STATUS`
    - Across all connections
  - `SHOW SESSION STATUS`
    - Just this connection
- Provides a report with over 250 metrics on server operation!

SERVER STATUS

`Connections`	Number of new connections established
`Max_used_connections`	Check if it matches max_connections
`Threads_cached`	Number of threads in “thread_cache”
`Threads_connected`	Number of concurrent connections
`Threads_created`	Threads created (thread cache misses)
`Threads_running`	Queries currently executing

SERVER STATUS

- `Open_tables`
  - Number of currently open tables
  - Single table opened twice counts as two
  - Check that table_cache is large enough to accommodate open_tables
- `Opened_table`
  - Number of times table was opened (table_cache miss)
  - Check how many opens per seconds are happening:

```
mysqladmin –i 1 –r extended-status | fgrep opened_table
```

SERVER STATUS

- `Slow_queries`
  - Queries considered to be slow (`long_query_time`)
  - Logged in slow query log if it is enabled (discussed in next lecture)

ENGINE STATUS

- `SHOW ENGINE InnoDB STATUS\G`
    - Great way to see what is going on with InnoDB!
    - File IO
    - Buffer Pool
    - Log Activity
    - Row Activity
    - Lock information, deadlocks, transaction status, pending operations, etc.

PERSISTENT VARIABLES

- To change one of the tuning variables permanently, simply put it in `my.cnf` and restart the server:
  - `table_cache = 128`
  - `max_connection = 200`
  - `...`
- Some of the many available tuning variables:

TUNING VARIABLES

- `max_connections`
  - Maximum number of connections to the server.
- `thread_cache_size`
  - Cache up to this many treads after disconnect.
- `table_cache`
  - Number of tables MySQL can keep open at the same time. Closing/ opening table is expensive, but it does eat RAM to keep too many open..

TUNING VARIABLES

- There are lots of variables out there (recall section 5.1.3), and fortunately, many, many sample configurations to consider - MySQL even ships with several configuration variations to use as templates.
- There are also literally hundreds of guidelines, white papers, blogs, books and forums to help with tweaking every little parameter.
- Many factors will go into the final configuration, including hardware setup, operating system, database size/use/features/demands and **testing**

REVERSE DNS

- Due to the flexibility of the ACL system, MySQL normally performs a reverse DNS lookup when a client connects, in order to determine the host name. But the lookup process can be slow, resulting in sluggish connection speeds.
- Unfortunately, these lookups are needed if grant tables use host names instead of IPs..
- So the best practice, if possible, is to use IPs, not host names, for MySQL accounts!
- To fully disable these reverse lookups:
    - Set `skip-name-resolve` in `my.cnf`, restart MySQL

LAB

1) Adjust the table cache to 128, thread cache to 32 and disable DNS reverse lookups.

2) Scroll through the status output for your server. Look up the meaning of at least 5 variables that are not immediately apparent to you.

slideshow.end();