MySQL & PHP for Web Applications

Peter Zaitsev,
MySQL AB

Open Source Revolution
2002-10-17

Contents

MySQL Background

- Founders of MySQL,Monty and David used **a lot** of OpenSource software when they started working on UNIX in the early 80s
- Richard Stallman talk in Stockholm 1996
- Needed a SQL Database for web pages
  - Technical reasons: tools was not suited for web use
  - Political reasons: customers wanted standards and accessibility from Windows (ODBC)

MySQL Background

- So they decided to develop their own commercial database inspired by OpenSource projects
- MySQL development started 1995 based on experience and code from Monty's and David's previous database

MySQL Ancient History

- 1996: MySQL 3.11 publicly released
- 1997: First commercial licenses and support agreements sold
- 2000: license changed to the GPL
  - Maximum spread of MySQL was the goal, so standard license was required

MySQL Modern History

- 2001: MySQL AB is being sued by NuSphere and is suing NuSphere for violating the GPL
- 2001: A single MySQL AB is formed
  - Lead investor ABN Amro one of europes largest investors
- 2002: Peter Joined MySQL
- 2002: Subsidiaries in Germany and USA Opened

Some MySQL Users

- NASA / US Census Bureu (Web sites)
- Yahoo Finance & 10+ other parts of Yahoo
- Slashdot.org, Freshmeat.net & Linux.com (Web)
- Nortel(InSight) / Ericsson / Alcatel / Telia / Nokia
- Motorola / Compaq (Embedded)
- Cisco (Embedded)
- Some famous names are not to be disclosed
- And thousands of other commercial customers

Some numbers

- # installations
  - Difficult to judge
  - Included in Linux distributions
  - 1 million+ copies has been shipped with MacOS X
  - Included with Solaris
- **>20000+** server downloads per **day**
- Overture.com (nr/max $)
  - MySQL 23/1.02
  - Oracle 50/6.80
  - Postgres 0/0.00

Google Rates MySQL

- Links according to Google.com
  - MySQL.com 31800
  - Oracle.com 16000
  - Postgresql.org 6130
- Mentions according to Google.com
  - MySQL 5040000
  - Oracle 6760000
  - PostgreSQL 2110000

Documentation

- MySQL has very good documentation as a online reference
- About 600 pages, updated as new code is written
  - *We write code and documentation at the same time (and we have done that from the start!)*
- A printed version of Manual is available
- MySQL AB now has a full time technical writer
- Excellent books from Paul Dubois
- A lot of third party books are also available in many different languages

MySQL as a Company

- Distributed company
- Head office in Uppsala, Sweden, Other (sales and admin) offices in USA, Finland and Germany
- Currently about 60 people in over 14 countries
- Developers mostly working from home

MySQL and Making Money

- We get this question all the time!
- By **Commercial Licences**
- By **Support** Agreements
- By **Partnership** Agreements
- By **Training** Courses
- By **Certification** (Beta version available **NOW**!)
- By Professional Services (**Consulting**)
- By **Advertising** on our website
- Place **your** orders online at order.mysql.com

MySQL Design Philosophy

- Modular design
- Never pay in performance for unused features
- Easy to install
- Easy and practical to use
- Only do solid implementations.
  - Do not add a hack just to have a feature (that almost works)
- Write documentation while writing code
- Occams razor: No complexity beyond necessity

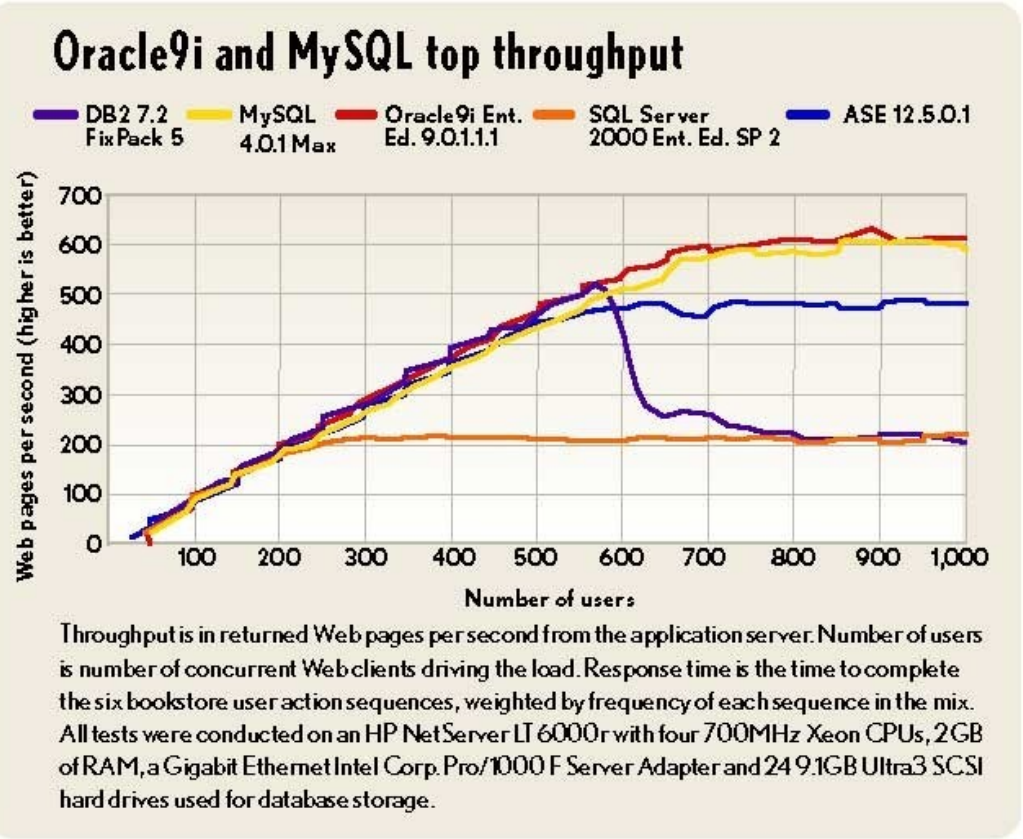
MySQL Features

- An extended subset of ANSI SQL89.
- Versioning ACID transaction support (InnoDB)
- Numerous extensions like SELECT .... LIMIT 10,10
- Few limits: disk, memory and OS set the limits
- Handles large datasets (hundreds of GB)
- Can be extended with User Defined Functions(UDF) in C
- Embeddable engine (MySQL server as a library)
- Full text search
- Support for > 4GB tables and easy BLOB support
- Master-Slave Replication for data redundancy
- Per table choice of backend (with/without transactions)

MySQL Platform support

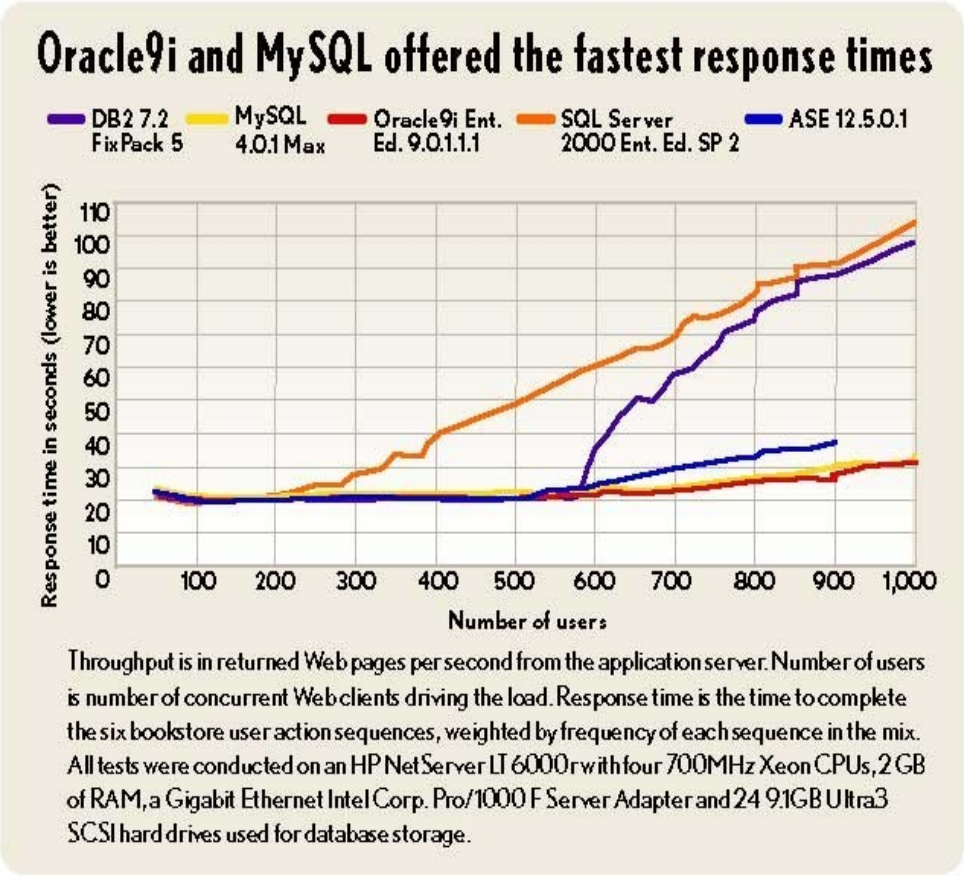
- Written to be easily ported and OpenSource so users can port it easily.
- Native (SMP) or Emulated, User level threads can be used
- 32bit and 64bit environment support
- Currently supported platforms: Solaris 2.4+, Linux 2.0+, HPUX 10.20+, AIX 4+, MacOS X, Tru64 Unix, BSDI 2.x+, FreeBSD 2.x+, OS/2, SCO UnixWare, IRIX 6.x, Windows 95+, NetBSD, OpenBSD 2.0+, SCO OpenServe, AmigaOS
- Port on a way: QNX, Novell

Speed in Web Applications



- First in 7 years full scale independend test of leading databases by Eweek
- Done in January 2002
- Tested MySQL 4.0, Oracle 9i, DB2 7.2, MS SQL Server 2000SP2 and Sybase 12.5

MySQL and Oracle are winers



- MySQL & Oracle both offer best througput responce times
- Test were done on MySQL unfriendly Windows platform but it still able to complete
- We're still working to improve performance

Some citatations from Eweek

- "The Oracle and MySQL drivers had the best combination of a complete JDBC feature set and stability."
- "Of the five databases we tested, only Oracle9i and MySQL were able to run our Nile application as originally written for 8 hours without problems."
- "SQL Server and MySQL were the easiest to tune, and Oracle9i was the most difficult because it has so many separate memory caches that can be adjusted."

MySQL Storage Engines

- Web application may greatly benefit from using appropriate table types for different tables
- A storage engine is a low level data storage / retrieval module (disk or memory)
- This allows you to choose locking and speed tradeoffs per table (instead of when choosing db!)
- MySQL supported multiple storage engines from the very beginning
- CREATE TABLE **TYPE=HEAP** (key int, value char(10), PRIMARY INDEX key);
- ALTER TABLE table_name **TYPE=InnoDB;**

MyISAM – Classic Storage Engine

- Developed by MySQL AB (replaced original ISAM)
- Static, dynamic and compressed (read-only) row formats but no transactions
- Text and compressed indexes
- Data and indexes in separate files
- Fast read/write performance but low r/w concurrency
- Extremely good concurrency in the select and insert at end case (logs)
- Needs to be checked/repaired in case of crash
- Especially useful for websites & logging
- Full text indexes support

INNODB – High Performance Transactions Engine

- Actively developed code from InnoBase Oy
- Full transactions (ACID) with **row** level locking
- Better concurrency than MyISAM for read/write
- Consistent reads (Oracle style MVCC)
- Uses tablespaces instead of individual files
- MySQL AB provides support for InnoDB through a contract with the creators/developers
- Is included in MySQL 4 & the MySQL Max binarys
- Has now been in active use under heavy load
- Fast and automatic crash recovery
- Good for storing sensitive ie financial information
- Foreign keys support

HEAP – In Memory table

- Developed by MySQL AB
- Completely in Memory with very fast **hash** based indexing
- Useful for
  - Temporary tables
  - Lookup tables
  - Summary tables which you can regenerate quickly
- Bad for range queries until MySQL 4.1
- No BLOB support until MySQL 4.1
- Content lost at server shutdown
- CREATE TEMPORARY TABLE a(...) type=HEAP;

BDB/BerkeleyDB tables

- Developed and supported by Sleepycat Software Inc
- Full transactions (ACID) with **page** level locking
- Better concurrency than MyISAM for read/write on the same table but worse than INNODB
- Primary key lookups can be faster than MyISAM
- Data and indexes in one file per table
- Is included in the MySQL Max binary
- Can also be used at a lower level independent of MySQL
- First transaction safe table handler in MySQL
- Most users prefer INNODB now

Replication. High Availibility and Scalibility

- MySQL supports many sites that need **high** reliability
  - *This is done by "mirroring" the data to many machines*
- The single **Master** logs SQL commands that update data
- **Slaves** connect to the master or another slave to read, and rerun the updates
- All Slaves can be used for select type queries increasing scalability
- Examples of users are
  - *Yahoo (Visit Jeremy Zawodny's talk Friday to learn everything!)*
  - *Mobile.de (sells used cars, 100+ M page views/month)*

MySQL 4.0

- Currently Beta Expected to be stable in 2-3 months
- Is used in production envinronment by many users, including Yahoo
- Improved replication
- Better integrated INNODB Tables
- Support for "UNION"
- Multi table DELETE
- QueryCache (3x boost for some web sites)
- Setting variable values online
- Low level HANDLER functions for more speed

MySQL 4.0 More features

- Boolean FULLTEXT search.
- SQL_CALC_ROWS/FOUND_ROWS() - use 1 query instead of 2 for data listing
- Cryptography functions: SHA1(), AES_ENCRYPT(), DES_ENCRYPT()
- OpenSSL for secure connections
- Limit user activity MAX_QUERIES_PER_HOUR etc
- Available as Embedded Server library

MySQL 4.0 Faster

- SELECT COUNT(DISTINCT ...) ...
- Bulk loading of data
- Bulk updates of full text indexes
- Removing all the rows in a table
  - *TRUNCATE TABLE table_name;*
- SELECT * FROM table WHERE blob_col like "%keyword%"
  - *Uses a fast turbo Boyer-More stringsearch*
- CREATE TABLE foo DATA DIRECTORY="/path/to/dir" INDEX DIRECTORY="/path/to/dir"
  - *Support for spreading MyISAM files over many disks*
- Multithreaded index rebuilding in myisamchk

MySQL 4.1 Features

- Available as pre-Alpha from BitKeeper Tree for testing
- Alpha release is planned in 2 months
- SELECT row1 FROM table1 WHERE a=(SELECT b FROM table2)
  - *Sub-Select. Yes, we know this is very late*
- SELECT * FROM table1, (SELECT b from table2) WHERE ...
- UPDATE t1,t2,t3 SET t1.c1=Val,t2.c2=val2 WHERE t1.id=t2.id and t2.id=t3.id
  - *Derived tables & Multiple table UPDATE*

MySQL 4.1 Features

- Multidimensional data support (OpenGIS)
  - *2d and 3d data*
  - *New point, line, multipoint, polygon etc datatypes*
- UNICODE support
- Online help for SQL syntax in the command line client
- Secure MySQL client-connections with SSL
- InMemory tables can use B-Tree keys (so they also can handle ranges queries well), support auto_increment columns and BLOBs
- More features to come during alpha development

MySQL 4.1 New Client Server Protocol

- Prepared Statement support
  - *Prepare SELECT foo from bar where a=?*
  - *Do ("1")*
  - *Do ("2")*
- Binary protocol to run faster
- No need to quote binary data
- More sequre authentication
- Done by Venu Anugati, San Fransisco, USA

MySQL 5.0

- Stored Procedures (we be starting with ANSI-99 Syntax)
  - *Project lead Per-Erik Martin in Uppsala, Sweden with two other full time developers backing him up*
- Referential Integrety (Foreign keys) for all table types
- Online backup of MyISAM tables
- New column types:
  - *BIT*
  - *True VARCHAR (no space trimming)*
  - *ARRAY*

MySQL 5.0

- Better concurrency for table cache and key cache
- New text based FRM format
- Expression default values for columns
- VIEWS
- Improved optimizer (OR handling etc)
- Cursors
- Warning/Info System
- **We need your wishes to add to the list**

Why to use PHP for Web Applications

- PHP is used by vast group of users and growthing
- PHP is easy to learn language
- PHP has Excellent MySQL support
- PHP has huge amount of modules developed
- PHP allows you to use OO style
- PHP was designed for Web Applications
  - *Is a Web Server module for best performance*
  - *Can be easily build into HTML*
  - *Uses high performance Zend Engine*

Web Application Platform

- These are the parts of Web Application Platform:
  - *System Hardware*
  - *Network Infrastructure*
  - *Operation System*
  - *Application envinronment*

Selecting Hardware

- Allmost everything you like - Both MySQL & PHP are highly portable
- MySQL & PHP benefit from SMP and HyperThreading technology
- 64bit CPU is good for large data sizes
- Cache size and memory/bus bandwith matters
- RAID & Multiple disk improve performance
- Use Reliable hardware
- AMD CPU's instead of Intel can offer more CPU power cheaper

Networking Issues

- Internal network as important as externel
- Use compatible network hardware
- 100Mbit or 1000Mbit network may improve performance
- Lantency is as important as bandwidth
- Check if network handles large load
- Multiple network cards could give benefit

Operation System

- Check status of Operation system for MySQL & PHP
- Try to avoid Exotic and Old Operation system
- Linux and Solaris are most widely used
- OS Should support your hardware fully
- Adjust parameters – defaults could be bad
- Which Linux flavour do we use ?
  - *SuSE Linux, RedHat is used by many users*
  - *Recent 2.4.x kernel from the vendor*
  - *ReiserFS and EXT3 are filesystems of choice*

Application Envinroment

- Apache Server is good choice in Most cases
- Apache 1.3 is still more widely supported
- Use Recent Version of Apache/PHP/MySQL
- Use MySQL binaries from the site if possible
- Test all new versions you install
- Do not upgrade unless
  - *You have problems*
  - *Critical for you bug is fixed*
  - *You belive you will benefit from the changes*

Arhitecture hints

- Several less powerfull servers are often better than one
- Use High Availability solutions
- Test if they work
- Split MySQL and Web server on different hosts
  - *Security reasons*
  - *More easy tuning*
  - *Better availability and performance*

Performance & High Availibility

- Use MySQL internal replication to reach both
- Split data based on some splitting rule
- May use low level replication (DRBD,NDB)
- LVM & similar can help with snapshots
- Use distributed file system or network file system to share the data
- Rsync can be used for periodic syncronisation
- LinuxVirtualServer or similar projects can be used to maintain Web Server HA
- Use monitoring software, ie MOODSS

Configuring for Performance

- Default configuration might be good, but not for your case.
- Advices good for one case might be bad for another
- Check if advice matches your application
- Benchmarking is the best way to find an answer

Configuring Operation System

- Check hardware drivers to use optimal mode.
  - *Write-Back for battary backup RAID*
  - *Full Duplex for Network*
- Configure Virtual Memory and File System
  - *Reduce Swapping tendency*
  - *Active file cache*
  - *Delayed writes*
  - *"noatime" is a good mode for filesystems*
- Adjust number of open files and processes for user MySQL is using
- Take a look at TCP/IP settings

Configuring Web Server

- Server side cache like OOPS behind could be good choice
- Use Different Web servers for Static (images) and Dynamic content to save memory
- Use fast (thttpd or in-kernel) HTTP server for static content
- Set appropriate TimeOut and KeepAlive settings to ballance between performance and memory consuption
- Adjust number of maximum allowed clients to block overload
- Monitor server performance for further ideas

Configure PHP

- Check you have all required extentions. Do not forget MySQL
- Set Appropriate max_execution_time
- Limit allowed amount of memory to match your needs
- Depending on the application you can reach better performance with or without persistent connections

MySQL configuration basics

- Do not run MySQL under **root** user
- Use one of provided configuration templates
- Decide which amount of memory you would like MySQL to consume
- Limit max_connections value
- Use large table_cache value – open files do not cost you much
- Use ps/top to check amout of memory used by MySQL as guidance.
- You can do this at runtime with MySQL 4.0
- Recommended values highly depends on the application

Fine Tuning MySQL

- Optimizing Schema and Queries gives more than Variable tuning
- Create benchmarking query set for your application
  - *You may use logging or super-smack for this*
- Enable slow query logging.
- Check slow query log for new queries during application development
- Check EXPLAIN output for such queries
- Read MySQL server status for finding bottlenecks
- "SHOW INNODB STATUS" and VMSTAT are also good tools

MySQL Status Analyses

- "mysqladmin extended" - nice way to get the status
- "mysqladmin -r -i5 extended" - Live monitoring
- Values to check:
  - *Com_** - query types server handles
  - *Created_tmp_disk_tables* - increase tmp_table_size
  - *Created_tmp_files* - increase sort_buffer_size
  - *Handler_read_rnd_next* - Table scans. Add indexes
  - *Key_read_requests/Key_reads* – Key cache hit ration
  - *Opened_tables* - increase table_cache
  - *Qcache_hits* - query cache efficiency

Typical Problems and Solutions

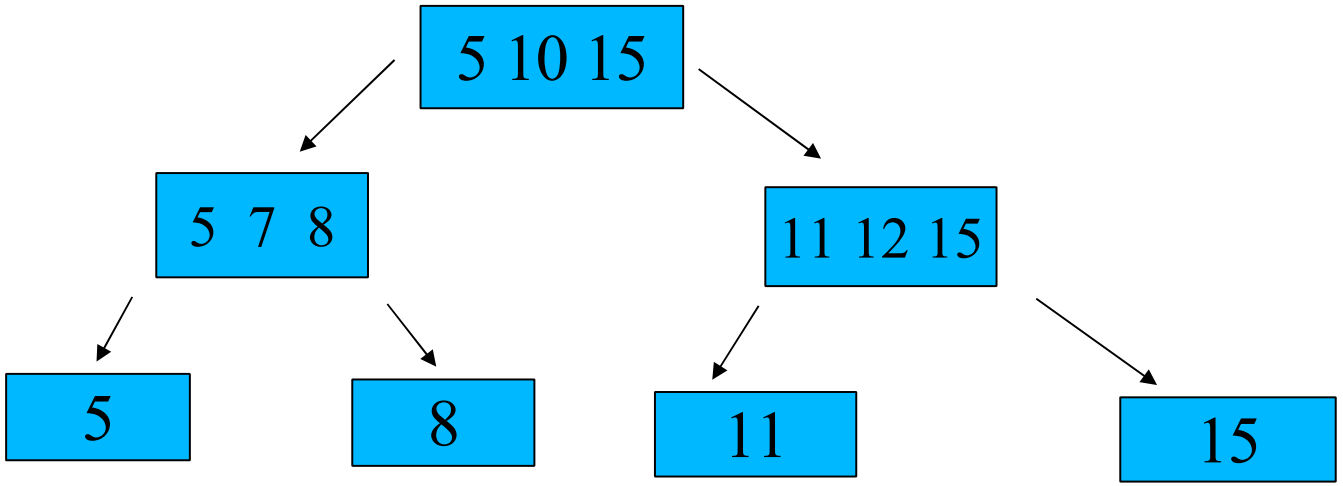
- Optimizing Search Applications
- Working with Tree type structures
- Result Set Navigation
- Optimizing queries with OR
- Timeouting Select Queries
- Session Management

Search Applications Optimization

- Do not use LIKE "%keyword%" queries
- FULLTEXT indexes are great
- May use Fulltext -> LIKE "%keyword%" fallback
- Manual fulltext indexing can be performed
- Searched data may be easily split
- For fixed number of queries – pregenerate results
- Query cache can give great benefit
- Temporary tables can be used as cache to avoid queries with different limit
- Use FOUND_ROWS() to avoid running two queries

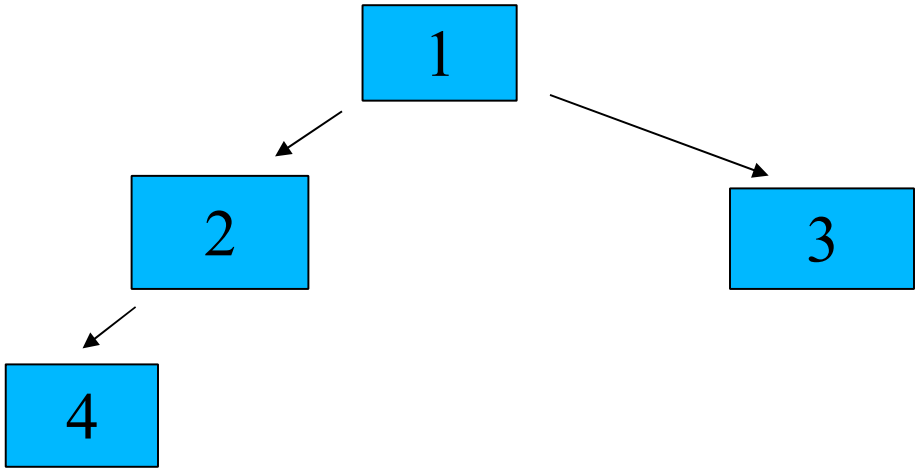
Tree Presentation #1

- Use special node numbering for constant trees for fast access:
- SELECT * FROM tree WHERE id between (<left> and <right>) will select whole subtree



Tree Presentation #2

- For dynamic tree use inheritance table
- Select * from parents where id=<N> order by level will select path to root
- Select * from parents where parent=N will select subtree



ID	LEVEL	PARENT
1	0	1
2	0	2
2	1	1
3	0	3
3	1	1
4	0	4
4	1	2
4	2	1

Result Set navigation

- For small result sets LIMIT is the simplest choice
- LIMIT 1000000,10 is slow
- If you can precalculate the order, do it and use "WHERE position between (1000000 and 1000010)" instead
- You can cache result for complex queries
  - *Run create table x select * from ... limit 1000 to cache first 1000 rows*
  - *Use this table as a cache to quickly return 10 requested rows to the client.*

Optimizing OR Queries

- Queries of "SELECT * FROM foo WHERE a="A" or b="B" is slow in MySQL (We plan to fix it)
- Prior to MySQL 4.0 you can use temporary heap tables to solve this – they are very fast in MySQL
- In MySQL 4.0 you can use UNION to resolve such queries if you do not have ORDER/GROUP BY clause
- SELECT * FROM foo WHERE a="A" UNION ALL SELECT * FROM foo WHERE b="b" and a<>"A" will have the same result

Query Timeouts

- Queries might be executing while user is already timed out
- No Internal functions to solve it yet.
- You can add comment to query: SELECT /* TIMEOUT 5 */ FROM foo;
- Use scripts which runs “mysqladmin processlist” periodically to spot and kill such queries

Session Management

- PHP default file based session handler can be very slow for busy sites.
- One may prefer MySQL based session handler because of using several Web Servers
- A lot of implementations already available
- Use HEAP tables if you can affort loosing of session data

Time for Questions

- More Information: http://www.mysql.com
- Documentation: http://www.mysql.com/doc
- Asking Question: mysql@lists.mysql.com

Enjoy MySQL !