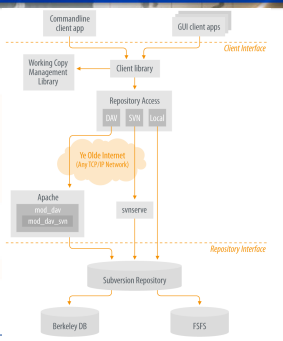
Make the most of Subversion 1.5

July 9th , 2008

SUBVERSION COLLABNET

Architecture Overview



Flexibility at three levels of the architecture:

- Multiple clients
  - Unix, Windows, Mac
  - Command-line vs. GUI
- Multiple ways to access the repository
- Multiple data stores available

SUBVERSION COLLABNET

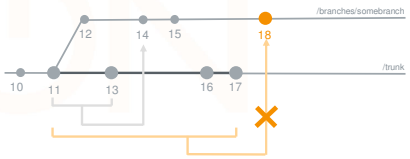
Subversion 1.5 - new features and their usage

- Merge tracking
- Sparse checkout
- Write/through proxy setup
- Change lists

www.collab.net COLLABNET

Merging Pre-1.5

- Merge is utility basically outside of Subversion core
- Responsibility to decide what needs to be merged is on the shoulder of the developer
- Quality issues can arise if the developer is not diligent



SUBVERSION — COLLABNET

Why Merge Tracking?

- Improves the usability
  - No more manual recordings of merges
  - Prevent unnecessary duplicate merges
  - svnmerge.py helps, but any merge outside of the script breaks the info
- Traceability
  - Recording what changes made it to a branch
  - The layout of the recording is now unified
  - Gives the capability to securely idetify the way of a change throughout the repository

SUBVERSION — COLLABNET

What is Merge Tracking in 1.5?

- Merge becomes part of Subversion vs. a utility
- It supports
  - Cherry picking of changesets
  - Documenting the manual merge of changesets
  - Preventing duplicate merges when repeatingly merging branches
  - Record manual merges outside of Subversion
  - Merge history - svn:mergeinfo answers merge related questions
- Supports activities when merging changes implemented by others

SUBVERSION — COLLABNET

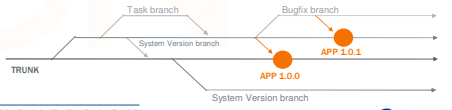
Pre-1.5 - Unstable Trunk model

- What's on a branch?
  - Trunk - ongoing development for next release
  - Release branches - formal promotion process
  - Isolated development branches - experimental work
- Use cases
  - Waterfall, serial development (OpenSource)
  - Single customer with smaller team

SUBVERSION COLLABNET

Useful for Stable Trunk Model

- What's on a branch?
  - Trunk - stable releases
  - Release - reflects work done for a release
  - Task - isolating longer tasks with instability
- Use Cases
  - Waterfall modell and overlapping development cycles
  - Larger teams



SUBVERSION COLLABNET

Better Support for Agile Release

- What's on a branch?
  - Trunk - stable points of releases and bugfixes
  - Release branch - release development process
  - Task branch - definable units of work
- Use cases
  - Agile development and release

SUBVERSION COLLABNET

Why does it matter?

- The introduction of merge tracking reduces the gap between Subversion and „High-end“ version control systems
- It enables the use of Subversion in more projects
  - Long maintenance cycles
  - Many different flavours of software (customer specific changes on a common core)
- This feature allows Philips Healthcare to rely on Subversion for the core platform project
  - Enables FDA compliance with a reasonable effort

SUBVERSION — COLLABNET

Simplified Audit Trail

- Tracks changes that are propagated through various branches
- If Subversion is deployed with an integrated tracker, issues are linked to revisions (changesets)
- Disclaimer: limitations of the 1.5 implementation
  - If the file structure differs heavily between 2 branches, potential conflicts might not be treated in the expected way
  - Example: a file has been moved in a branch, but modified in another branch
  - Visit the Subversion project to review tree conflict work going on

SUBVERSION — COLLABNET

Sparse Checkouts

- The problem: unnecessary data in your working copy
  - Modules you never edit
  - Large files that are kept along with source code (for example, build results, documents)
  - Sibling projects in the same repository
- Sparse checkouts give you the capability to opt-in/opt-out parts of the repository tree
  - Based on a directory tree depth selection
  - Selected depth is „sticky“ and maintained on the following operations

SUBVERSION — COLLABNET

Sparse Checkouts

- Adds a new –depth option
  - .. to svn checkout: selects the initial working copy depth
  - .. to svn update: modify a previous selection
  - .. to most of the other subcommands
    - Universilly supercedes the –recurse and –non-recurse options
    - Limits the scope of operations
- Does not require a Subversion 1.5 server!
  - When requesting depths from older servers, the client does the filtering to remove out-of-scope data
  - A server upgrade is still recommended to improve the performance

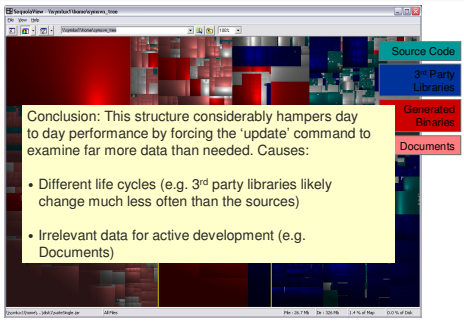
Improve the Workspace Management

- Minimizes the update, creation and switch time of workspaces
  - Speeds up day-to-day operations
- Minimizing the resource utilization on the clients
- Taking away the time pressure to optimize „grown“ repositories
  - Pre-1.5 you had to re-org your repository if the structure was not well planned

Good time to review the repository structure

- Distiguish groups to examine update frequency during common activities (e.g. regular development, testing,maintenance)
  - e.g. source code, 3rd party libraries etc.
- Use a tool like SequoiaView to create a graphic overview
  - Exclude the .svn directories when you run it on a working copy
  - Taylored color coding helps to identify the categorie you are interested in

Analysis - Real-life Example



Conclusion: This structure considerably hampers day to day performance by forcing the 'update' command to examine far more data than needed. Causes:

- Different life cycles (e.g. 3rd party libraries likely change much less often than the sources)
- Irrelevant data for active development (e.g. Documents)

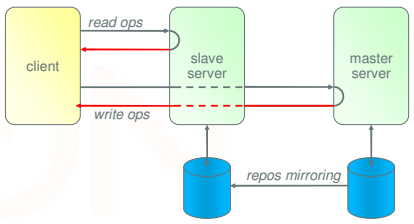
SUBVERSION — COLLABNET

WebDAV Write-through proxies

- Allows Subversion servers to pass data changing activities to another server
- Enables singlemaster/many slave server repository setups
  - Clients interact 90% of the time with a local slave server to cover the read activities
  - All write activities (commit, property changes, lock/unlock)are passed through to the master repository
  - The master propagates any changes back to the slave servers
  - Replication and re-sync is done with svnsync (introduced in 1.4)

SUBVERSION — COLLABNET

WebDAV Write-thru Proxies



SUBVERSION — COLLABNET

Some Use Cases

- Using mirrors gives you an online mirror to improve high-availability
  - In case of a server failure user can still work read-only immediately
  - Extends the time slot to correct the server problem
- Using mirrors helps with a backup strategy
  - The online mirror can be kept up-to-date immediately (via hook script)
  - The dump of a mirror can be used to recreate the master

SUBVERSION — COLLABNET

Work Distributed

- It is a different approach as the traditional replication model
- Subversion has been optimized to live in the internet
- Based on the experiences the performance is often a perceived problem
  - Consider when dealing with large files and limited bandwidth
  - Try the central approach first
  - Balance performance increase vs. infrastructure overhead

SUBVERSION — COLLABNET

Change list

- Changelists are a new client only feature
- Groups files to limit the scope of operations (e.g., log)
- Use a seeding concept for working copies to share changelists
  - A working copy with a baseline is created on a server
  - Authentication information is stored outside of the working copy
  - Simple copy and re-use
- Example: Subversion user from the financial sector improved compliance by eliminating errors during commit

SUBVERSION — COLLABNET

Summary

- Subversion 1.5 provides a lot of new capabilities that improve day-to-day activities
- Evaluate before rolling out
  - Define clear rules (i.e. naming conventions)
- Ensure that developers get trained

SUBVERSION — COLLABNET

Summary

Questions?!

SUBVERSION — COLLABNET