Search

all pages recent changes

# Internals/OptimizeHTTP

## Optimizing a repository for HTTP transfer

To reduce number of files needed to transfer over network, optimize –http command packs a repository into two tarballs, basic.tar.gz and patches.tar.gz, with following content:

### basic.tar.gz

1. _darcs/hashed_inventory
2. _darcs/meta-filelist-pristine
3. _darcs/meta-filelist-inventories
4. _darcs/meta-*
5. _darcs/hashed.pristine/*
6. _darcs/inventories/*

meta-filelist-* files contain directory listings for hashed.pristine and inventories dirs, in reverse order wrt tarball itself. While getting, files from this listings are downloaded using cache in parallel with tarball.

meta-* files in general contain additional files and information that could extend the tarballs functionality in some way. They are expected to have a small size, so that negative effect on performance would be minimal.

### patches.tar.gz

1. _darcs/patches/*

## Getting an optimized repository

1. Download and unpack basic.tar.gz. Result: lazy repository from time when optimize –http has been done.
2. Pull from parent repository. Result: lazy repository from current time.

3. Download and unpack patches.tar.gz. Result: full repository.

## Benchmarks

How does “optimize –http” improve the user experience?

- Jérémie’s repo (~900 patches): from 10s (get –no-packs) to 1s (get)
- `http://darcs.net/` (http://darcs.net/) (~9300 patches): “darcs optimize –http” takes 14s to run. _darcs goes from 54 MBytes to 64 MBytes (indeed _darcs/packs/ is 11 MBytes) Complete get: from 37 to 2 minutes, lazy get from 27 seconds to 7 seconds.

screened + 12 patches:

| . | packs | no-packs |
|---|---|---|
| lazy | 30s | 1m30 |
| full | 2m30s | 31m |