OBSIDIAN

OBSIDIAN 0.1 DOCUMENTATION

Clement Game
clement(at)digi-nation(dot)com

Keywords: Peer-to-Peer,Fast Rplication,mirroring,Dexp,Linux

Contents

1 Introduction

This is the documentation of the Obsidian Peer-to-Peer mirroring Software, version 0.1. This paper is released under CC-by-SA licence.

In ancient times, People was using a volcanic stone, which amount of

2 Overview

Obsidian is a fast-mirroring, Peer-to-Peer software which was specially designed to maintain large file repositories synchronized with each others, featuring very fast changes propagating.

To achieve this, Obsidian use 2 core technologies: Filesystem Events Listening , and a new protocol specially designed for the occasion: DEXP, acronym for Documents EXchange Protocol. With DEXP, Change Notifications are sent directly to the mirroring peers, so they don't have to poll periodically for file changes.

To illustrate how fast is Obsidian compared to a classical multiple rsync-based system, we can do the math:

Let's consider a chain of n peers, each node being synchronized to the next with a periodical rsynch check (T_c).The time taken by each node to download a change is T_d. In this case, the worst-case propagating time will be:

$T_r = \sum_0^n T_d(n) + T_c(n)$

As we can see in this case (rsync) , the more nodes we have, the more time it loses, because of the accumulation the T_c components.

But in the other case (Obsidian) , the calculation of the maximum propagating time between n nodes is:

$T_r = \sum_0^n T_d(n)$

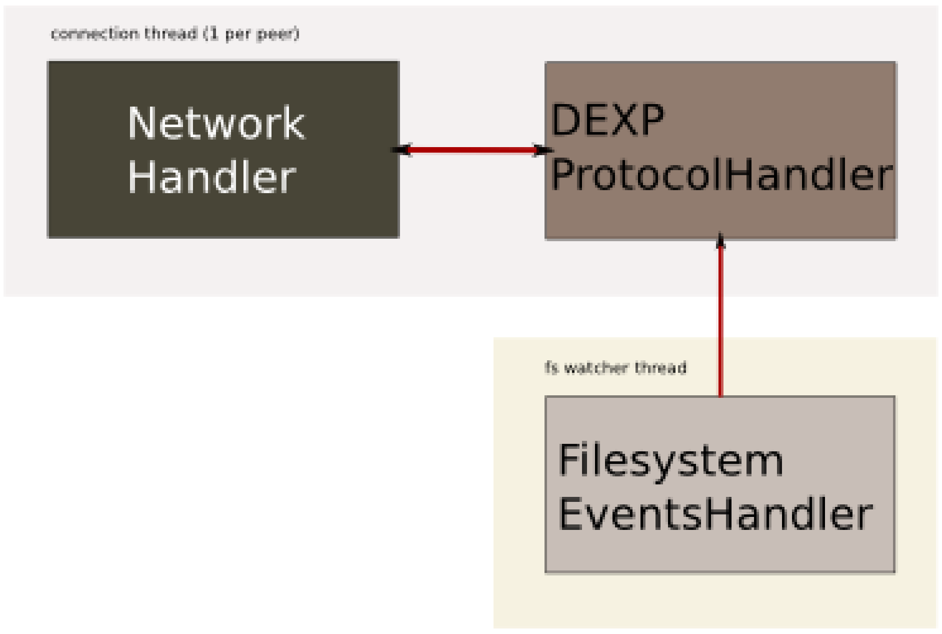
So with obisidan, The more nodes you haves, the more efficient it is compared to a periodic-poll based method.If we now consider a T_c constant for each node, we can evaluate the speed gain with:
$T_g = nT_c$
So the bigger is n, the bigger is the gain.

3 Global Architecture

This section explains

4 Software Architecture

The Software architecture behind Obsidian is quite easy, and fit as the following:



5 Getting Started with Obsidian

This section describes in detail how to get started with Obsidian, step-by-step and from scratch.

5.1 prerequisites

In order to build and run Obsidian, you'll need:

- A GNU/Linux Oprating System, with kernel 2.6.19 or higher.
- The OpenSSL Library plus its headers (libssl-dev) , 0.9.8 or higher.
- Cross-platform make (cmake 2.6 or higher).
- The GNU C compiler (gcc-4.X).
- Git DVCS(to retrieve the source code, if not done yet)

5.2 Fetching the sources

Here we're assuming that that you haven't fetched the obsidian source code yet. To do that, you will need the GIT DVCS, a well-known .
Once you have git installed on your system, just type:

```
git clone git://github.com/digination/obsidian.git
```

5.3 Compilation

This step explains how to build Obsidian from sources

5.4 Installation

To install Obsidian, Nothing is more simple. From the Obsidian Root Directory, just type ”make install” , and the Makefile will take care of the rest.

5.5 Configuration

6 Advanced Usages

6.1 Setting up TLS on your obsidian Server

6.2 Running multiple sessions of Obsidian on the same server

In order to circumvence the zero-dir limitation, you can choose to run multiple instances of obsidian at the same time on your server, so your peers can replicate data from the multiple instances.

7 More on DEXP

8 Known Bugs, Limitations

9 Special Thanks