

# Un petit début avec OpenDHT

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 31 mai 2007. Dernière mise à jour le 15 mai 2009

<https://www.bortzmeyer.org/opendht-debut.html>

---

Les DHT, les **tables de hachage distribuées** ont de plus en plus de succès pour stocker et récupérer des informations sur Internet. OpenDHT était un service que tout le monde pouvait utiliser sans avoir à installer sa propre DHT.

Il existe au moins trois mises en œuvre de DHT en logiciel libre, Chord, Pastry et Bamboo <<http://www.bamboo-dht.org>>. Mais créer une DHT nécessite de les installer sur un certain nombre de machines et tout le monde n'a pas forcément accès à un réseau comme celui de PlanetLab. D'où l'idée toute simple de OpenDHT : installer une DHT (en l'occurrence Bamboo) et permettre à tous de l'utiliser via une simple interface XML-RPC.

À titre d'exemple, voici deux petits scripts Python, un qui enregistre l'adresse IP de ma machine dans la DHT et l'autre qui la récupère. Cela permet, par exemple, des systèmes de rendez-vous entre des machines dont l'adresse IP est variable. Voici un exemple d'utilisation (le script ajoute le préfixe HOST-REGISTRATION pour limiter les risques de confusion avec une autre application) :

```
% ./register-address.py
I am "munzer.bortzmeyer.org" (208.75.84.80)
Registration: Success

% ./get.py HOST-REGISTRATION-munzer.bortzmeyer.org
208.75.84.80
```

Voici le code de `register-address.py` :

```

import socket
import xmlrpclib
import sha

# Seconds
ttl = 3600
gateway = "http://opendht.nyuld.net:5851/"
secret = "foo-bar"

host_info = socket.gethostbyaddr(socket.gethostname())
name = host_info[0]
addresses = host_info[2]
address = addresses[0]

print "I am \"%s\" (%s)" % (name, address)

proxy = xmlrpclib.ServerProxy(gateway)
results = {0:"Success", 1:"Capacity", 2:"Again"}
key = xmlrpclib.Binary(sha.new("HOST-REGISTRATION-%s" % name).digest())
val = xmlrpclib.Binary(address)
shash = xmlrpclib.Binary(sha.new(secret).digest())
result = proxy.put_removable(key, val, "SHA", shash, ttl,
                             "register-address.py")
print "Registration: %s" % results[result]

```

Notons que le code pour découvrir l'adresse IP locale (`socket.gethostbyaddr(socket.gethostname())`) est très limité. Notamment, si la machine est derrière un routeur NAT, l'adresse récupérée sera probablement une adresse privée, pas celle qui est vue sur l'Internet. Il faudrait modifier ce programme pour utiliser un protocole de découverte d'adresse comme STUN (RFC 5389<sup>1</sup>).

Et voici le code de `get.py` :

```

import sys
import sha
import optparse
from xmlrpclib import *

p = optparse.OptionParser(usage="usage: %prog [options] <key>")
p.add_option("-g", "--gateway", dest="gateway", metavar="GW",
             default="http://opendht.nyuld.net:5851/",
             help="gateway URI, list at http://opendht.org/servers.txt")
p.add_option("-d", "--details", dest="details", action="store_true",
             help="print secret hash and TTL remaining for each value")
p.add_option("-m", "--maxvals", dest="max", default="10", metavar="CNT",
             type="int", help="how many values to return")
(opts, args) = p.parse_args()
if (len(args) < 1): p.print_help(); sys.exit(1)
pxy = ServerProxy(opts.gateway); maxvals = int(opts.max)
pm = Binary(""); key = Binary(sha.new(args[0]).digest())
while 1:
    if (opts.details):
        vals, pm = pxy.get_details(key, maxvals, pm, "get.py")
        for v in vals:
            hex = '0x' + ''.join(["%02x"%ord(x) for x in v[3].data[:4]])
            print v[0].data, v[1], v[2], hex
    else:
        vals, pm = pxy.get(key, maxvals, pm, "get.py")
        if not vals:

```

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5389.txt>

```
    print "No result"
else:
    for v in vals:
        print v.data
if (pm.data == ""): break
```

OpenDHT, service expérimental, a toujours été peu fiable. Les pannes étaient fréquentes et les clients voulant utiliser OpenDHT en vrai devaient être robustes. Le 3 mai 2009, le responsable, Sean Rhea a annoncé qu'il arrêterait la maintenance, et stoppait OpenDHT le 1er juillet. Un successeur possible est OpenLookup <<http://openlookup.net/>>.