

BitDew User Guide

Installation, Usage and Programming
Edition 1, for BitDew version 0.2.3
August 18 2009

Gilles Fedak & Haiwu He

Copyright © 2006, 2007, 2008, 2009 by Institut National de Recherche en Informatique (<http://www.inria.fr/>).

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the author.

1 Overview

1.1 What is BitDew ?

The BitDew framework is a programmable environment for management and distribution of data on computational Desktop Grids.

BitDew is a subsystem which can be easily integrated into Desktop Grid systems (XtremWeb, BOINC, Condor etc..). Currently, Desktop Grids are mostly limited to embarrassingly parallel applications with few data dependencies. BitDew objective is to broaden the use of Desktop Grids. Our approach is to break the "data wall" by providing in single package the key P2P technologies (DHT, BitTorrent) and high level programming interfaces. We first target Desktop Grid with peta-scale data system : up to 1K files/nodes, with size up to 1GB and distributed to 10K to 100K nodes.

The BitDew framework will enable the support for data-intense parameter sweep applications, long-running applications which requires distributed checkpoint services, workflow applications and maybe in the future soft-realtime and stream processing applications.

1.2 What Can I do with BitDew ?

BitDew offers programmers a simple API for creating, accessing, storing and moving data with ease, even on highly dynamic and volatile environments.

The BitDew programming model relies on 5 abstractions to manage the data : i) replication indicates how many occurrences of a data should be available at the same time on the network, ii) fault-tolerance controls the policy in presence of machine crash, iii) lifetime is an attribute absolute or relative to the existence of other data, which decides the life cycle of a data in the system, iv) affinity drives movement of data according to dependency rules, v) protocol gives the runtime environment hints about the protocol to distribute the data (http, ftp or bittorrent). Programmers define for every data these simple criteria, and let the BitDew runtime environment manage operations of data creation, deletion, movement, replication, and fault-tolerance operation.

1.3 Bitdew Architecture

The BitDew runtime environment is a flexible environment implementing the API. It relies both on centralized and distributed protocols for indexing, storage and transfers providing reliability, scalability and high-performance.

The architecture follows a classical three-tiers schema commonly found in Desktop Grids: it divides the world in two sets of nodes : stable nodes and volatile nodes. Stable nodes run various independent services which compose the runtime environment: Data Repository (DR), Data Catalog (DC), Data Transfer (DT) and Data Scheduler (DC). We call these nodes the service hosts. Volatile nodes can either ask for storage resources (we call them client hosts) or offer their local storage (they are called reservoir hosts). Usually, programmers will not use directly the various D* services; instead they will use the API which in turn hides the complexity of internal protocols.

The Bitdew runtime environment delegates a large number of operation to third party components : 1) Meta-data information are serialized using a traditional SQL database, 2)

data transfer are realized out-of-band by specialized file transfer protocols and 3) publish and look-up of data replica is enabled by the means of DHT protocols. One feature of the system is that all of these components can be replaced and plugged-in by the users, allowing them to select the most adequate subsystem according to their own criteria like performance, reliability and scalability.

2 Downloading and Compiling BitDew

2.1 Downloading BitDew

The BitDew files are downloadable from the [BitDew web site](http://www.bitdew.net) (<http://www.bitdew.net>):

`'bitdew-stand-alone-0.2.3.jar'`

contains everything you need to run the software.

`'bitdew-src-0.2.3.jar'`

contains the sources of BitDew as well as librairies needed to compile BitDew.

`'bitdew-lib-0.2.3.jar'`

contains the BitDew classes.

2.2 Compiling BitDew

2.2.1 The Short Way

Deflate the distribution with the command :

```
unzip bitdew-src-0.2.3.zip
```

You don't need to compile BitDew, unless you have modified the sources. To compile BitDew, move in the BitDew directory and execute the following commande :

```
build.sh
```

First, the command preprocesses `'idl'` files, from which it generates `'java'`. Next, it invokes the java compiler and places `'class'` files in the `'lib'` directory.

The following command generates everything you need :

```
build.sh release
```

2.2.2 Advanced Usage

When developping BitDew, you may need to use the following build targets as well :

```
build.sh clean
```

Delete files generated by the build process.

```
build.sh tests
```

Run the Junit unitary tests.

```
build.sh tests-reports
```

Same as `build.sh tests`, but results of unitary tests are placed in the `'reports'` directory.

```
build.sh javadoc
```

Build the javadoc for the various APIs.

```
build.sh srcdoc
```

Build the full source code documentation. This targets depends on the Doxygen tool. It also requires time, processing power and disk space.

```
build.sh jar
```

Create a new `'bitdew-0.2.3.jar'` file.

`build.sh stand-alone-jar`

Create a new 'bitdew-stand-alone-0.2.3.jar' file.

`build.sh release`

Make a new BitDew release, that is, run most of the previous targets.

3 Running BitDew

3.1 Quickstart

The simplest way is to use the file ‘bitdew-stand-alone-0.2.3.jar’ which contains all the files and libraries included in one single jar file.

You can launch the command-line tool simply with the following command, which will display the usage :

```
java -jar bitdew-stand-alone-0.2.3.jar
```

and to obtain the complete list of options :

```
java -jar bitdew-stand-alone-0.2.3.jar --help
```

The tool can either start services or act as a client.

To start all of the services supported by BitDew simply run the two following commands :

```
java -jar bitdew-stand-alone-0.2.3.jar serv dc dr dt ds
```

This will start the following services dc : Data Catalog, dr : Data Repository, dt : Data Transfer and ds: Data Scheduling.

```
java -jar bitdew-stand-alone-0.2.3.jar serv dc dr dt ds
```

3.2 Invoking the command line tool

The format for running the BitDew command line program is:

```
java -jar bitdew-stand-alone-0.2.3.jar options commands command options ...
```

If the command line seems too long to type for you, we recommend to set an alias in your ‘.bashrc’ as this :

```
alias bitdew="java -jar bitdew-stand-alone-0.2.3.jar "
```

BitDew supports the following options, shown by the output of `java -jar bitdew-stand-alone-0.2.3.jar --help`:

BitDew version 0.0.1

BitDew command line client

Usage : java -jar bitdew-stand-alone.jar [Options] Commands [Command Options]

Options:

-h, --help	display this helps
-d, --dir	working directory
--host	service hostname
--port	service port

Services:

serv [dc dr dt ds]	start the list of services separated by a space
--------------------	---

Attributes:

attr attr_definition	create attribute where attr_definition has the syntax att_Name = field1=value1, field2=value2.
----------------------	--

1

<pre> replicat=int affinity=dataId lftabs=int lftabs=dataId oob=protocol ft=[true false] </pre>	<p>Field can have the following values :</p> <p>number of data replicat in the system. The special value -1 means that the data will be replicated to each node</p> <p>affinity to data Identifier. Schedule the data on node where dataId is present.</p> <p>absolute life time. The value is the life duration in minutes.</p> <p>relative lifetime. The data will be obsolete when dataId is deleted.</p> <p>out-of-band file transfer protocol. Protocol can be one of the following [dummy ftp bittorrent]</p> <p>fault tolerance. If true data will be rescheduled if one host holding the data is considered as dead.</p>
---	--

4 Index

C

compilation 3

D

distribution 3

download 3

F

file manifest 3

G

getting help 5

H

help 5

I

invoking 5

O

options 5

U

usage 5

V

version 5

5 Table of contents

Table of Contents

1	Overview	1
1.1	What is BitDew ?	1
1.2	What Can I do with BitDew ?	1
1.3	Bitdew Architecture	1
2	Downloading and Compiling BitDew	3
2.1	Downloading BitDew	3
2.2	Compiling BitDew	3
2.2.1	The Short Way	3
2.2.2	Advanced Usage	3
3	Running BitDew	5
3.1	Quickstart	5
3.2	Invoking the command line tool	5
4	Index	7
5	Table of contents	8