BitDew User Guide

Installation, Usage and Programming
Edition 1, for BitDew version 1.0
July 13 2012

Gilles Fedak & Haiwu He

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 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`):

'`bitdew-stand-alone-1.0.jar`'
: contains everything you need to run the software.

'`bitdew-src-1.0.jar`'
: contains the sources of BitDew as well as librairies needed to compile BitDew.

'`bitdew-lib-1.0.jar`'
: contains the BitDew classes.

2.2 Compiling BitDew

2.2.1 The Short Way

Deflate the distribution with the command :

```
unzip bitdew-src-1.0.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-1.0.jar`' file.

`build.sh stand-alone-jar`
: Create a new '`bitdew-stand-alone-1.0.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-1.0.jar`' which contains all the files and libraries inclued in one signle jar file.

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

```
java -jar bitdew-stand-alone-1.0.jar
```

and to obtain the complete list of options :

```
java -jar bitdew-stand-alone-1.0.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-1.0.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-1.0.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-1.0.jar` *`options commands command options ...`*

If the command line seems too long to type for you, we recommand to set an alias in your '`.bashrc`' as this :

```
alias bitdew="java -jar  bitdew-stand-alone-1.0.jar "
```

BitDew supports the following options, shown by the output of `java -jar bitdew-stand-alone-1.0.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.
                                Field can have the following values :
```

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

4 Index

5 Table of contents

Table of Contents