Development: How can I install the latest Python in Bright?

RedHat, SUSE, and similar Linux-based OS distributors provide versions of Python that lag behind the latest upstream. This is normally a good thing, since the distributors provide integration, and carry out testing to make sure it works well with the rest of the OS. However, some admins would like to have the latest Python versions available on their cluster. One reason may be that later versions have some nicer features.

Upgrading to the latest Python outside of what the distributors provide is not supported by the Linux distributors. So, it is possible to run into Python-related issues for which support cannot be provided by the distributors, and also not by us. Bright will of course help out within reason.

That said, upgrade-related issues have not been noticed. So there is a good chance that all works well.

The following steps were tested on a Bright 6.1 installation originally running Python 2.6. The steps are a guide to upgrading and recompiling Bright PythonCM with Python 3.4:

1. Download, extract, build Python 3.4.0:

```
# wget -c https://www.python.org/ftp/python/3.4.0/Python-3.4.0.tgz

# tar -xzvf Python-3.4.0.tgz

# cd Python-3.4.0

# ./configure --prefix=/opt/python/3.4.0 --enable-unicode=ucs4 --enable-shared

# make

# make install
```

2. Download, extract, build boost 1.55.0:

```
# wget -c http://sourceforge.net/projects/boost/files/boost/1.55.0/boost_1_55_0.tar.bz2

# tar -xjvf boost_1_55_0.tar.bz2

# cd boost_1_55_0

# echo "using python : 3.4 : /opt/python/3.4.0 :

/opt/python/3.4.0/include/python3.4m : /opt/python/3.4.0/lib ;" >>
./tools/build/v2/user-config.jam

# ./bootstrap.sh --with-python=/opt/python/3.4.0/bin/python3 --with-python-version=3.4
--with-python-root=/opt/python/3.4.0

# ./b2 --prefix=/opt/boost/1.55.0 --without-mpi --enable-unicode=ucs4 install
```

3. Extract the attached pythoncm archive and build it

```
# yumdownloader pythoncm-dist
```

```
# rpm2cpio pythoncm-dist-6.1-r20472_cm6.1.x86_64.rpm | cpio -idmv

# tar -xjvf pythoncm-6.1-r20472.tar.bz2

# cd pythoncm-6.1-r20472



# ./build.sh --with-boost=/opt/boost/1.55.0
--with-python-include=/opt/python/3.4.0/include/python3.4m
--with-python-lib=/opt/python/3.4.0/lib/libpython3.so



[...]

Compiled pythoncm revision: 20472

====================================

You have successfully built python

bindings for Bright Cluster Manager

====================================
```

Listing of files

build.sh:

```
#!/bin/bash

#

# This is the Bright Computing Python Cluster Manager bindings build script

#



exec > >(tee pythoncm.src.build.log)

exec 2> >(tee pythoncm.src.build.err)



base=`dirname $0`

if [ "$base" = "." ]; then

  base=`pwd`
```

```
fi

echo $base



function on_exit() {

  if [ $? == 0 -a -f "$base/python/pythoncm.so" ]; then

    export LD_LIBRARY_PATH="/opt/python/3.4.0/lib:./remotecm"

    echo "import sys" > pytest.py

    echo "sys.path.insert(0, 'python')" >> pytest.py

    echo "import pythoncm" >> pytest.py

    echo "clustermanager = pythoncm.ClusterManager()" >> pytest.py

    echo "print ('Compiled pythoncm revision: %s' % clustermanager.revision)" >> pytest.py

    /opt/python/3.4.0/bin/python3 pytest.py

    if [ $? == 0 ]; then

      echo "====================================="

      echo "You have successfully built python"

      echo "bindings for Bright Cluster Manager"

      echo "====================================="

    else

      echo "====================================="

      echo "You have compiled pythoncm,"

      echo "but it was unable to execute properly"

      echo "====================================="

    fi

  else

    echo "====================================="

    echo "pythoncm build failed!"
```

```
    echo "Check out the files:"

    echo "  pythoncm.src.build.log"

    echo "  pythoncm.src.build.err"

    echo "If the problem is not caused by missing"

    echo "packages (see README file) please send"

    echo "a mail to support@brightcomputing.com"

    echo "and attach both pythoncm.src.build file"

    echo "====================================="

  fi

}



set -e

trap on_exit EXIT



CWD=`pwd`

CPUS=`grep -Eic "^processor" /proc/cpuinfo`

if [ -z "$CPUS" ]; then

  CPUS=1

fi



# When compiling with a different version of pyhton make sure to also change the python
pytest.py command above

autoconf

./configure --with-gzip $*

make -j $CPUS
```

Unique solution ID: #1226
Author: Frank Furter
Last update: 2014-09-02 14:40