

BuildBot

OASIS4 Workshop 01.09.2010

Walter Sauf

Max-Planck-Institut for Meteorology
Hamburg

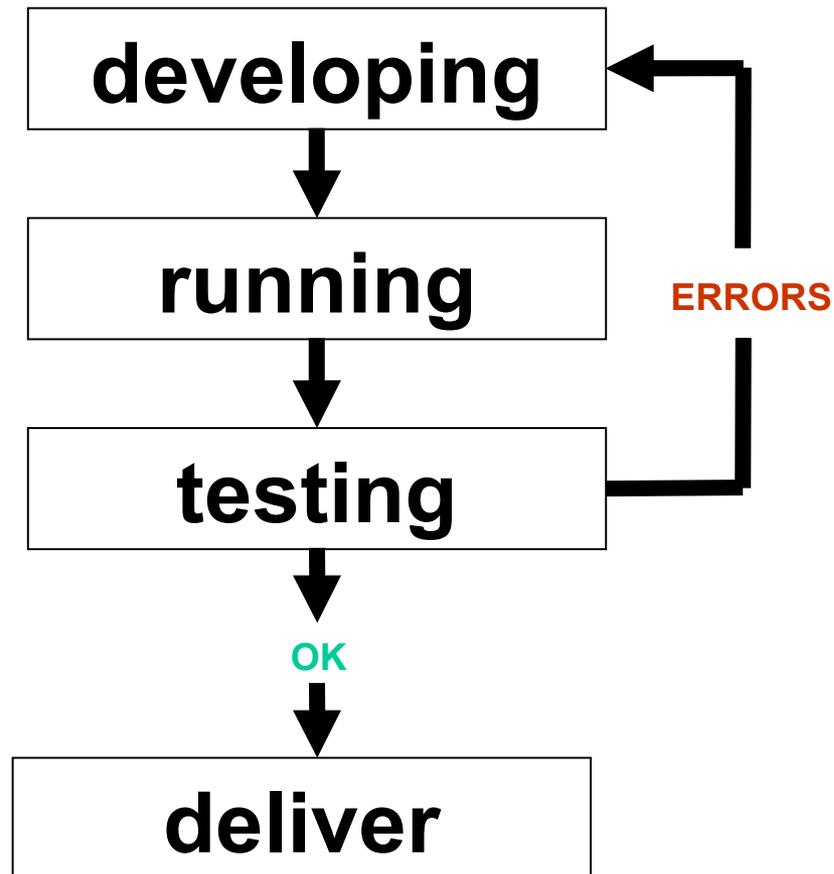


Overview

1. Motivation
2. What is BuildBot
3. Configuration BuildBot
4. BuildBots WebPages
5. Trigger a builder manual
6. Example ICON
7. Problems / Outlook

1. Motivation

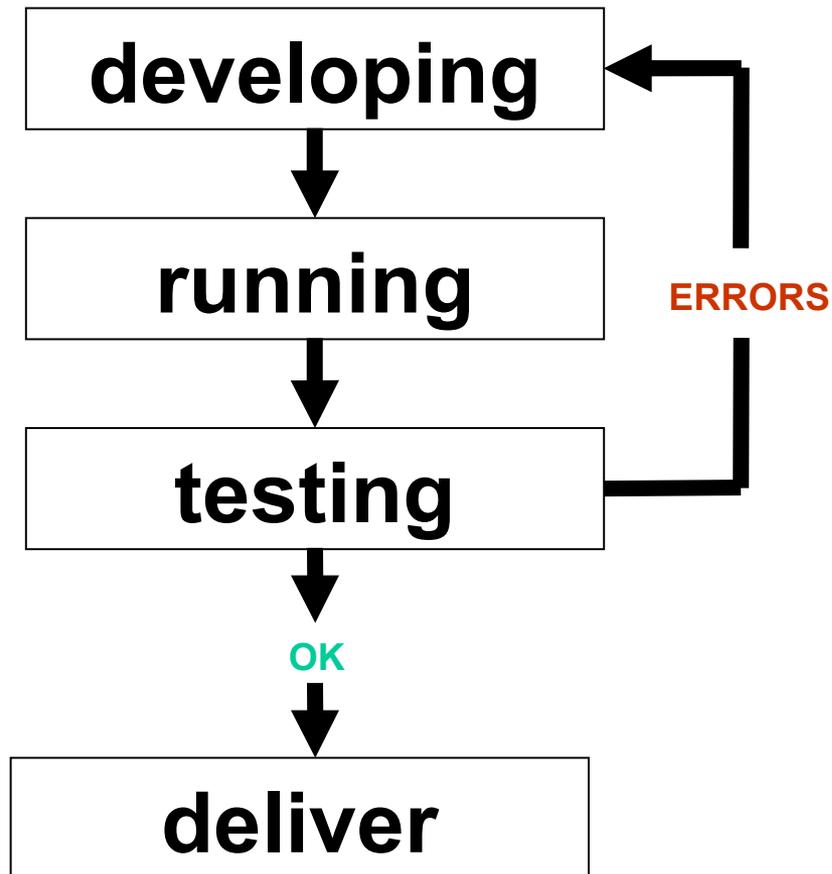
Building a new Program



1. Motivation

Building a new Program

Did I really
test enough ?



Questions

- Did I test the dependency of different compilers?
- Did I test the dependency of different systems?
- Do I get equal results using the old and new program version

Nice to have

- A tool which automatically checks the last developed version
- It should test the program on different systems
- It should test the program using different compilers or compiler options
- It should give feedback if the last test was successful or not.
- It should be easy to use.

Options

- Write a test-scripts and start it as a cronjob
- Write a program which automatically test the new versions
- Use a program like BuildBot which is developed for this job

2. What is BuildBot

It can automatically

- run a program on different computer systems.
- start a program with different parameters or program options
- test a program using different compilers or compiler-options
- compare results of different Model runs.

How does BuildBot work?

- It is divided into a server and several clients.
- Only one configuration file is needed to control a project.
- It provides a web-server to view the activity and results of the clients.
- It is written in python and should run on every system which has python (2.4.4) installed
- It is a free software project and is free of charge
- The latest version is [buildbot-0.8.1](#)



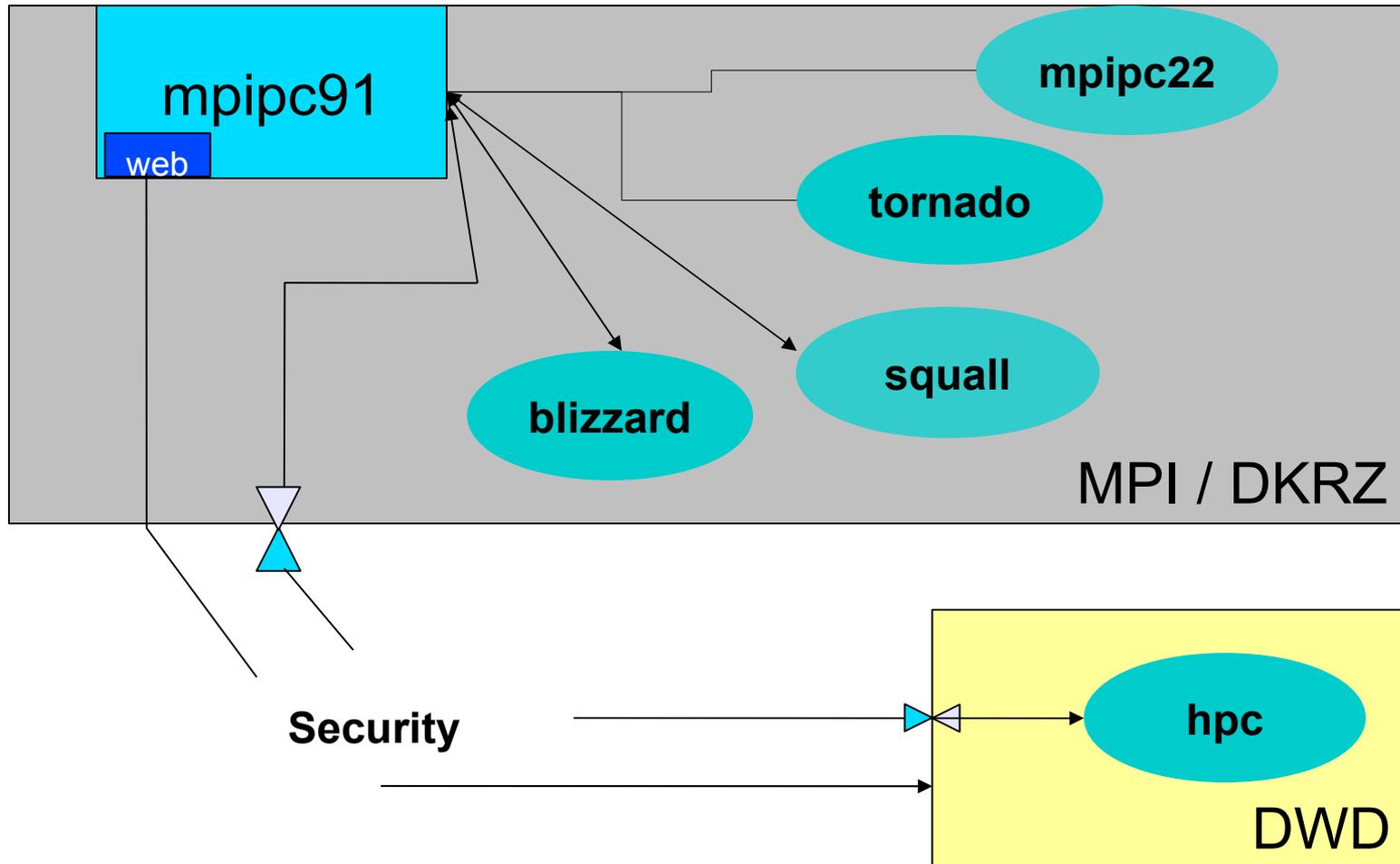
How does BuildBot work?

- On each client several **builders** can run
- It can run under a “normal” user account?
- It needs one port for the communication and one for the web server

BuildBot at the MPI

- We are using version buildbot-0.7.12.
- It is installed on blizzard, tornado, squall and a Linux-PC
- A connection to the DWD is planed

BuildBot



BuildBot

It can start

- **builders** at a special time, like each night.
- **builders** at each change of a repository.
- **builders** by triggering through the commando *buildbot* with *sendchange*.
- **builders** by triggering through the commando *buildbot* with *debugclient*.
- **builders** by triggering through the a button on the website

BuildBot

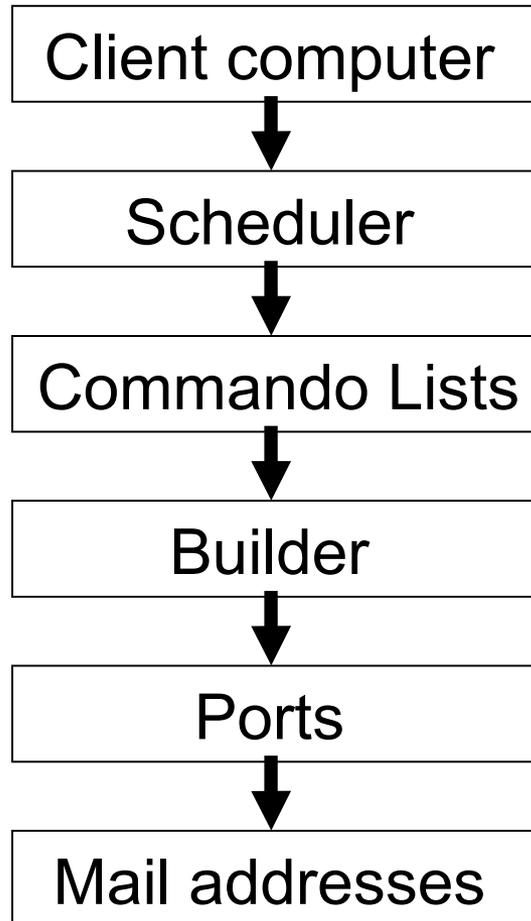
- Can start special revisions.
- Can run locally.
- Each **builder** runs in a separate directory.
- The Server sends the commandos to the clients
- Each builder sends its logging infomation back to the Server

What is a **builder** doing?

- It can checkout a version from repository (SVN, CVS)
- It can call system calls
- It can set special environment variables for each **builder**
- If an error occurs in a **builder** it interrupts the execution of the commando list.
- It can stop a commando when there is no response after a wait time (deadlock)

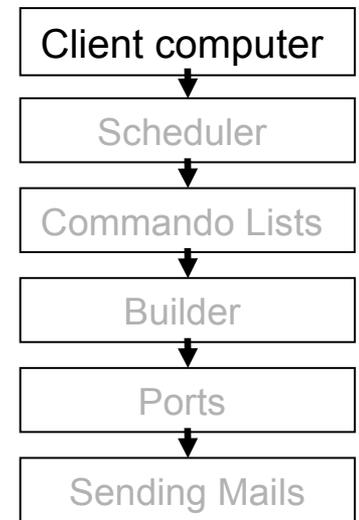
3. Configuration BuildBot

Definition of



Definition of client computer

```
c['slaves'] = [  
    BuildSlave("mpipc", "bot1keyword", max_builds=1),  
    BuildSlave("blizzard", "bot1keyword"),  
    BuildSlave("tornado", "bot1keyword"),  
    BuildSlave("squall", "bot1keyword")  
]  
)
```

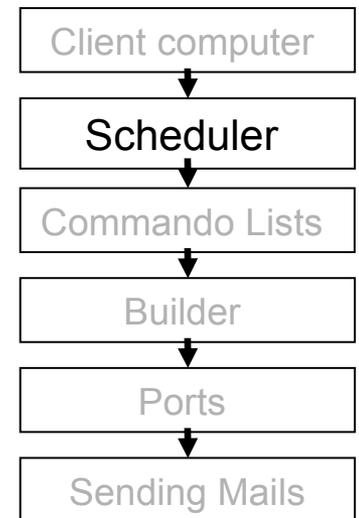


Definition of scheduler

```
s1 = AnyBranchScheduler('main',['icon-dev'],1*30,  
    [ 'MPIPC_nag', 'MPIPC_gcc','BLIZZARD',  
    'TORNADO_nag', 'TORNADO_gcc',  
    'SQUALL_nag', 'SQUALL_gcc'  
    ]
```

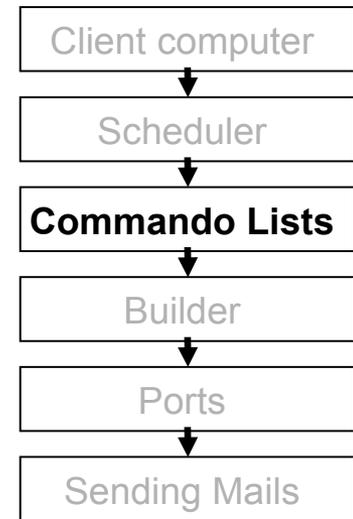
```
nightly = scheduler.Nightly(name='nightly',  
    builderNames=[  
    'MPIPC_nag', 'MPIPC_intel','BLIZZARD',  
    'TORNADO_nag', 'TORNADO_intel',  
    'SQUALL_nag','SQUALL_intel'  
    ],  
    branch='icon-dev',hour=1,minute=0)
```

```
c['schedulers'] = [s1,nightly]
```



Commando List for diff. Sources

```
m1 = factory.BuildFactory()
m1.addStep(ShellCommand(command='rm -rf *'))
m1.addStep(source.SVN(mode='update',
    baseURL='http://svn.zmaw.de/svn/p1/trunk/',
    defaultBranch='prog'))
m1.addStep(ShellCommand(command='./configure',
    haltOnFailure=True,timeout=3600))
m1.addStep(ShellCommand(command='make',haltOnFailure=True,timeout=3600))
m1.addStep(ShellCommand(command='run',haltOnFailure=True,timeout=36000))
#-----
m2 = factory.BuildFactory()
m2.addStep(ShellCommand(command='rm -rf *'))
m2.addStep(source.SVN(mode='update',
    baseURL='http://svn.zmaw.de/svn/p1/branch/',
    defaultBranch='prog'))
m2.addStep(ShellCommand(command='./configure',
    haltOnFailure=True,timeout=3600))
m2.addStep(ShellCommand(command='make',haltOnFailure=True,timeout=3600))
m2.addStep(ShellCommand(command='run',haltOnFailure=True,timeout=36000))
```

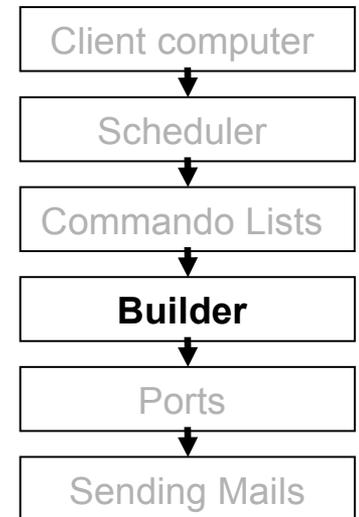


Problems and Solutions for Commando lists

- Transfer of information to the next calls
 - Read or write the information to a local file
- Loading of modules
 - Calling the “module load” in .profile
- Return Status in Scripts
 - Checking and returning the STATUS after each call
- Calling of *qsub* or *lsubmit* without sync
 - Using “-sync y” (qsub) or –s (lsubmit)

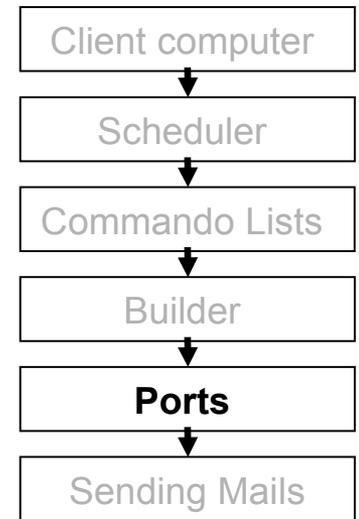
Builder

```
c['builders'] = [  
    { 'name'           : 'MPIPC_nag',           # builder name  
      'slavename'     : 'mpipc',               # Computer  
      'builddir'      : 'mpipc_nag',          # Builder working  
      'factory'       : 'm1',                 # Commando list  
      'env': {'BB_SLAVE': 'slave_1'}         # Env. Variable  
    },  
    ..  
    { 'name'           : 'BLIZZARD',  
      'slavename'     : 'blizzard',  
      'builddir'      : 'blizzard_1',  
      'factory'       : 'm2',  
      'env': {'BB_SYSTEM': 'blizzard', 'BB_SLAVE': 'slave_1'}  
    }  
  ]
```



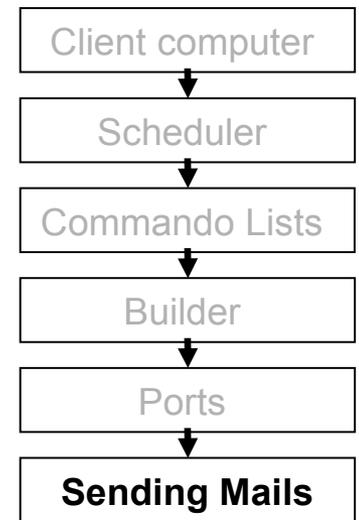
Ports

```
..  
..  
c['slavePortnum'] = 9988  
..  
..  
c['debugPassword'] = "debugKeyword"  
..  
..  
c['projectName'] = "ICON"  
c['projectURL'] = "http://icon.enes.org/"  
..  
c['buildbotURL'] = http://mpipc91.mpi.zmaw.de:8011/
```



Sending Mail

```
c['status'].append(mail.MailNotifier(  
    fromaddr="BUILDBOT_ICON<sender@zmaw.de>",  
    extraRecipients=[  
        "first.add@zmaw.de"  
        , "second.adr@zmaw.de"  
        , "third.add@zmaw.de"  
    ],  
    mode='failing',  
    subject='[BUILDBOT ICON] %(builder)s BUILD FAILED',  
    relayhost="smtp.zmaw.de",  
    sendToInterestedUsers=False  
)
```



“buildbot.tac”

Server:

```
..  
..  
basedir = r'/scratch/local1/m211098/master_ICON'  
configfile = r'ICON.cfg'  
..  
..
```

Client:

```
..  
basedir = r'/scratch/local2/buildbot/ICON'  
buildmaster_host = 'mpipc91.mpi.zmaw.de'  
port = 9988  
slavename = 'mpipc'  
passwd = 'bot1keyword'  
....  
umask = 022  
..
```

Start of Daemons

- To start the daemons the file *buildbot.tac* is needed
- To start the daemon call
buildbot start <server directory>
- All output info is written to a logging file
- No root permission is needed

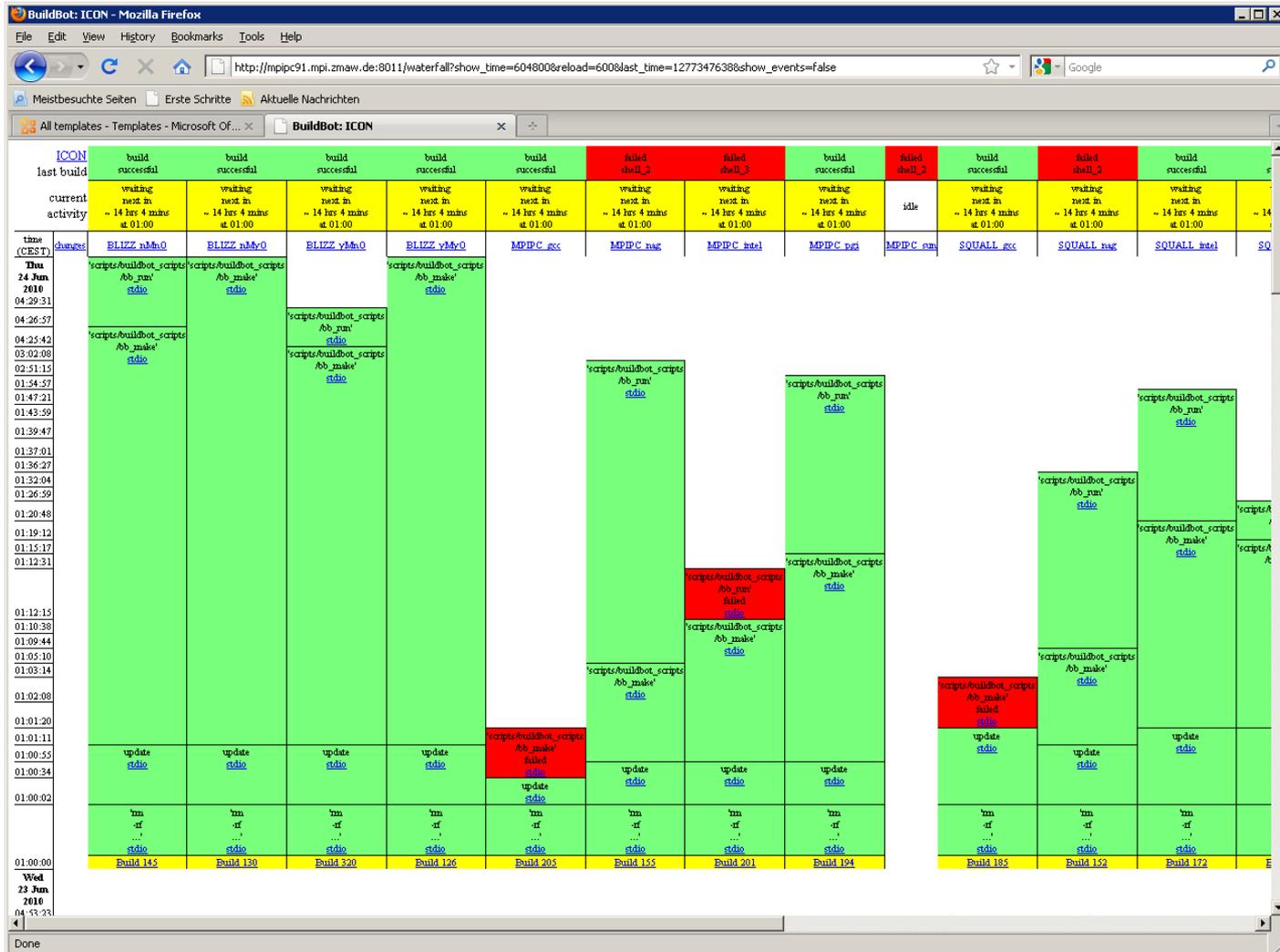
4. BuildBots WebPages

<http://mpipc91.mpi.zmaw.de:8010>

Welcome to the Buildbot!

- the [Waterfall Display](#) will give you a time-oriented summary of recent buildbot activity.
- the [Grid Display](#) will give you a developer-oriented summary of recent buildbot activity.
- The [Latest Build](#) for each builder is here.
- [Recent Builds](#) are summarized here, one per line.
- [Buildslave](#) information
- [ChangeSource](#) information.
- [About this Buildbot](#)

Waterfall Display



Waterfall Display (Zoom)

BuildBot: ICON - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://mpipc91.mpi.zmaw.de:8011/waterfall?show_time=604800&reload

Meistbesuchte Seiten Erste Schritte Aktuelle Nachrichten

All templates - Templates - Microsoft Of... x BuildBot: ICON x

		build successful	build successful	build successful	build successful	build successful
last build		successful	successful	successful	successful	successful
current activity		waiting next in ~ 14 hrs 4 mins at 01:00	waiting next in ~ 14 hrs 4 mins at 01:00	waiting next in ~ 14 hrs 4 mins at 01:00	waiting next in ~ 14 hrs 4 mins at 01:00	waiting next in ~ 14 hrs 4 mins at 01:00
time (CEST)	changes	BLIZZ rMfn0	BLIZZ rMfy0	BLIZZ yMfn0	BLIZZ yMfy0	MPPIC gcc
Thu 24 Jun 2010 04:29:31		'scripts/buildbot_scripts /bb_run' stdio	'scripts/buildbot_scripts /bb_make' stdio		'scripts/buildbot_scripts /bb_make' stdio	
04:26:57				'scripts/buildbot_scripts /bb_run' stdio		
04:25:42		'scripts/buildbot_scripts /bb_make' stdio				
03:02:08				'scripts/buildbot_scripts /bb_make' stdio		
02:51:15						
01:54:57						
01:47:21						
01:43:59						
01:39:47						
01:37:01						
01:36:27						
01:32:04						
01:26:59						

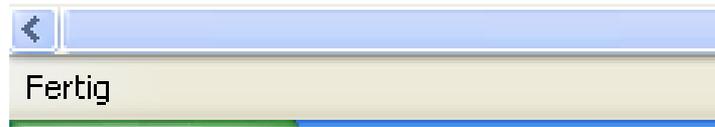
Waterfall Display : Parameter



- **reload=600**
Makes a automatic page updated every 10 minute.
- **show_events=false**
Disabled the showing of starting and stopping of the master and slaves
- **show_time=604800**
Is the time of the history shown. (maximal history = 14 days)

Waterfall Display

[\[next page\]](#) [\[help\]](#) [\[welcome\]](#) [\[Stop Reloading\]](#)
[Buildbot-0.7.10p1](#) working for the [ICON](#) project.
Page built: Mon 12 Jul 2010 18:43:56

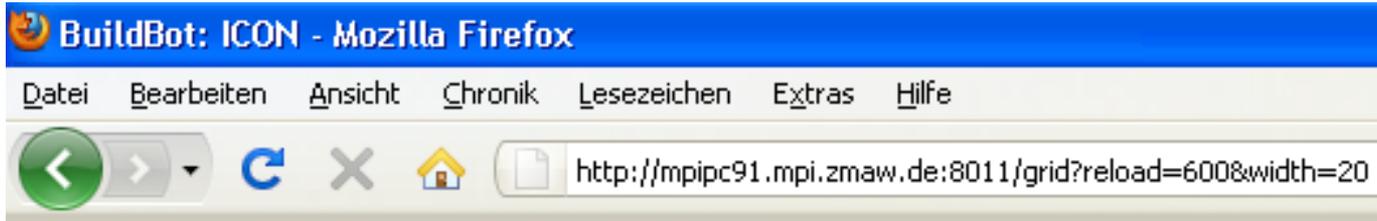


- Selecting the link “next page” shows former runs.

Grid Display

ICON	1868	1871	1876	1885	1888	1893	1900		1901	1904			1906				
	in 'icon-dev'	in 'icon-dev'	in 'icon-dev'	in 'icon-dev'	in 'icon-dev'	in 'icon-dev'	1900	1893	in 'icon-dev'	in 'icon-dev'	1902	1903	1904	in 'icon-dev'	1905	1906	in 'icon-dev'
BLIZZ_nMn0 (waiting)	OK	OK	OK	OK	OK	OK	failed shell_2		failed shell_2	failed shell_2				failed shell_2		OK	OK
BLIZZ_nMy0 (waiting)	OK	OK	OK	OK	OK	OK			failed shell_2	failed shell_2				failed shell_2		OK	OK
BLIZZ_yMn0 (waiting)	OK	OK	OK	OK	OK	OK			failed shell_2	failed shell_2				failed shell_2		OK	OK
BLIZZ_yMy0 (waiting)	OK	OK	OK	OK	OK	OK			failed shell_2	failed shell_2				failed shell_2		OK	OK
MPiPC_gcc (waiting)	OK	OK	OK	failed shell_2	failed shell_2	failed shell_3	failed shell_2	OK	failed shell_2	OK				OK		OK	OK
MPiPC_intel (waiting)	failed shell_3	failed shell_3	failed shell_3	failed shell_3	failed shell_3	failed shell_3			failed shell_2	failed shell_2				failed shell_2		failed shell_3	failed shell_3
MPiPC_nag (waiting)	OK	OK	OK	OK	OK	failed shell_3			failed shell_2	OK				OK		OK	OK
MPiPC_pgi (waiting)	OK	OK	OK	OK	OK	failed shell_3			failed shell_2	failed shell_2				failed shell_2		OK	OK
MPiPC_sun																failed shell_2	
SQUALL_gcc (building)	failed shell_3	OK	OK	failed shell_2	failed shell_2	OK			failed shell_2	failed shell_3	failed shell_3	failed shell_3	OK	OK		OK	OK
SQUALL_intel (waiting)	failed shell_3	OK	OK	OK	OK	OK			failed shell_2	failed shell_2			failed shell_2	failed shell_2		OK	failed shell_3
SQUALL_nag (waiting)	failed shell_3	OK	OK	OK	OK	OK			failed shell_2	failed shell_3	failed shell_3	failed shell_3	OK	OK		OK	failed shell_3
SQUALL_pgi (waiting)	failed shell_3	OK	OK	OK	OK	OK			failed shell_2	failed shell_2			failed shell_2	failed shell_2		OK	OK

Grid Display : Parameter



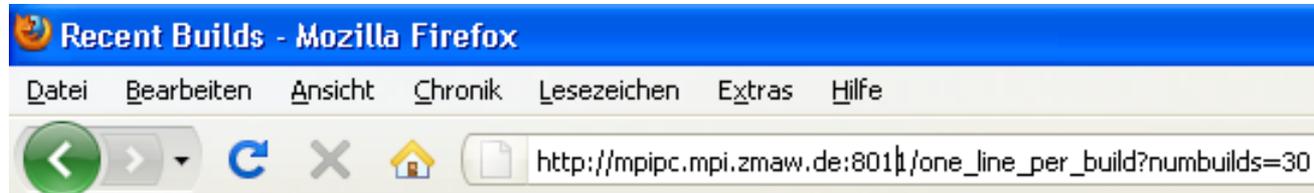
- **reload=600**
Makes a automatic page updated every 10 minute.
- **width=20**
Shows the last 20 runs

Recent Builds

Last 30 finished builds:

- (Jul 12 11:48) rev=[1940] **failure** [SQUALL_gcc #214](#): failed failed slave lost
- (Jul 12 01:00) rev=[1937] **failure** [SQUALL_intel #195](#): failed shell_3
- (Jul 12 01:00) rev=[1937] **failure** [SQUALL_pgi #188](#): failed shell_3
- (Jul 12 01:00) rev=[1937] **failure** [SQUALL_nag #180](#): failed shell_3
- (Jul 12 01:00) rev=[1937] **failure** [SQUALL_gcc #213](#): failed shell_3
- (Jul 12 01:00) rev=[1937] **success** [BLIZZ_yMyO #145](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [BLIZZ_nMyO #149](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [BLIZZ_nMnO #165](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [BLIZZ_yMnO #339](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [MPIPC_nag #176](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [MPIPC_pgi #214](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [MPIPC_gcc #230](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [TORNADO_sun #304](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [TORNADO_intel #181](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [TORNADO_pgi #175](#): build successful
- (Jul 12 01:00) rev=[1937] **success** [TORNADO_nag #172](#): build successful

Recent Builds: Parameter



numbuilds=30

Defines the number of shown builds

Buildslave

Build Slaves

1. [blizzard](#):

- Used by Builders: [BLIZZARD](#)
- Slave is currently connected
- Admin: Walter Sauf walter.sauf -at- zmaw.de
- Last heard from: about 16 hours ago (2010-Feb-01 02:31:04)
- Slave is idle.

2. [mpipc](#):

- Used by Builders: [MPIPC_nag](#), [MPIPC_gcc](#), [MPIPC_pgi](#), [MPIPC_sun](#), [MPIPC_intel](#)
- Slave is currently connected
- Admin: Walter Sauf walter.sauf -at- zmaw.de
- Last heard from: about 3 hours ago (2010-Feb-01 15:19:13)
- Slave is idle.

3. [squall](#):

- Used by Builders: [SQUALL_nag](#), [SQUALL_gcc](#), [SQUALL_pgi](#), [SQUALL_sun](#), [SQUALL_intel](#)
- Slave is currently connected
- Last heard from: about 15 hours ago (2010-Feb-01 03:24:34)
- Slave is idle.

4. [tornado](#):

- Used by Builders: [TORNADO_nag](#), [TORNADO_gcc](#), [TORNADO_pgi](#), [TORNADO_sun](#), [TORNADO_intel](#)
- **Slave is NOT currently connected**

Builder

[Builder MPIPC_intel](#): Build #12

Results:

failed shell_3

SourceStamp:

- Branch: icon-dev
- Got Revision: 1368

Buildslave:

[mpipc](#)

Reason:

The Nightly scheduler named 'nightly' triggered this build

Steps and Logfiles:

1. [shell](#) ['rm -rf ...']
 1. [stdio](#)
2. [svn](#) [update]
 1. [stdio](#)
3. [shell_2](#) ['scripts/buildbot_scripts/bb_make']
 1. [stdio](#)
4. [shell_3](#) ['scripts/buildbot_scripts/bb_run' failed]
 1. [stdio](#)

Build Properties:

Name	Value	Source
branch	icon-dev	Build
buildername	MPIPC_intel	Build
buildnumber	12	Build
got_revision	1368	Source
revision	None	Build
scheduler	nightly	Scheduler

Logging

- Each commando writes the logging info to a file called “stdio”
- The first line of the file “stdio” contains the commando
- The next information shows all environment variables.
- After the list of environment variables the output of the commando is written

Logging

- By using qsub or llsbmit the output file is cat into the log file.

5. Trigger a builder manual

- Using *sendchange*

```
buildbot sendchange \
```

```
--master mpipc91.mpi.zmaw.de:9988 \
```

```
-u name \
```

```
-b icon-dev \
```

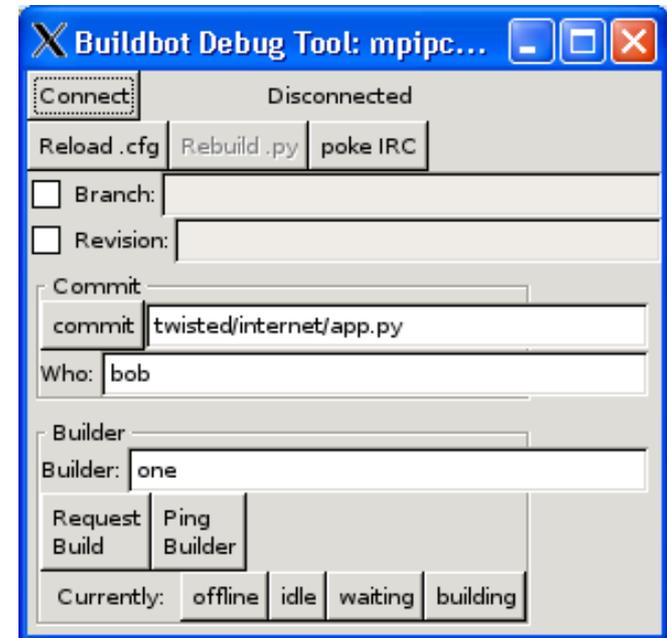
```
[-r 1234] \
```

```
"Testing of bb_run.new"
```

Trigger a builder manual

- Using the Debug Tool

```
buildbot debugclient \  
-m mpipc91.mpi.zmaw.de:9988 \  
-p debugkeyword
```



Trigger a builder manual

- Webpage
 - Single build
 - *allowForce=True*

d	failed
_2	shell_3
:	building ETA in ~ 9 mins at 18:05 next in ~ 7 hrs 4 mins at 01:00
<u>_sun</u>	<u>SQUALL_gcc</u>
	scripts/buildbot_scripts

Trigger a builder manual

Buildbot: SQUALL_gcc - Mozilla Firefox

http://127.0.0.1:8081/builders/SQUALL_gcc

Buildbot: SQUALL_gcc

[ICON](#)

Builder: SQUALL_gcc

Currently Building:

- [#214](#) ETA 547s (18:12:30) [shell_3]

Recent Builds:

- (Jul 12 01:00) rev=[1937] **failure** [#213](#): failed shell_3
- (Jul 11 01:00) rev=[1937] **failure** [#212](#): failed shell_3
- (Jul 10 01:00) rev=[1937] **failure** [#211](#): failed shell_3
- (Jul 09 05:09) rev=[1932] **failure** [#210](#): failed shell_3
- (Jul 08 17:02) rev=[1931] **failure** [#209](#): failed shell_3

Buildslaves:

1. [squal](#): CONNECTED
Admin: Walter Sauf walter.sauf@zmaw.de
Host info:
Host Beschreibung
HOSTNAME: squal11.zmaw.de
SYSTEM: Linux
KERNEL: 2.6.22-lustre-1.6.4.3-1
WORK-DIR /scratch/mpi/mh0287/

To force a build, fill out the following fields and push the 'Force Build' button

Your name:

Reason for build:

Branch to build:

Revision to build:

Currently Building:

- [#214](#) ETA 547s (18:12:30) [shell_3]

To force a build, fill out the following fields and push the 'Force Build' button

Your name:

Reason for build:

Branch to build:

Revision to build:

Trigger a builder manual

– All Builds

- On the main webpage select „*Latest build:*“
- At the end of the page you can see:

To force a build on all Builders, fill out the following fields and push the 'Force Build' button

Your name:

Reason for build:

Branch to build:

Revision to build:

6. Example ICON

What is a ICON **builder** doing?

1. It deletes the previous used files and directories
2. It checks out the last developed version of the project
- 3. It starts the script *bb_make***
- 4. It starts the script *bb_run***
5. If an error occurs in the build it interrupts the execution of the script.

Commando List for ICON

```
m1 = factory.BuildFactory()
m1.addStep(ShellCommand(command='rm -rf *'))
m1.addStep(source.SVN(mode='update'
baseURL='http://svn.zmaw.de/svn/icon/trunk/',
defaultBranch='icon-dev'))
m1.addStep(ShellCommand(
command='scripts/buildbot_scripts/bb_make',
haltOnFailure=True,timeout=36000))
m1.addStep(ShellCommand(
command='scripts/buildbot_scripts/bb_run',
haltOnFailure=True,timeout=36000))
```



ICON Builders

```
c['builders'] = [  
    {'name'          : 'MPIPC_nag',  
     'slavename'    : 'mpipc',  
     'builddir'     : 'mpipc_nag',  
     'factory'      : m1 ,  
     'env': {'BB_SYSTEM': 'mpipc', 'BB_SLAVE': 'mpipc_nag'}}  
    ],  
..  
    {'name'          : 'BLIZZ_nMnO',  
     'slavename'    : 'blizzard',  
     'builddir'     : 'blizz_nMnO',  
     'factory'      : m1 ,  
     'env': {'BB_SYSTEM': 'blizzard', 'BB_SLAVE': 'blizz_nMnO'}}  
    ]
```

Scripts

- Both scripts *bb_make* and *bb_run* are included in the repository.
- The control of system and compiler is done by analyzing the variables **BB_SYSTEM** and **BB_SLAVE**

bb_make

- The script creates the model binary
- It loads the needed Modules
- *qsub* and *lsubmit* are not used
- It calls the parts
 - configure
 - make
 - make index

Environment variables for BB_SLAVE

	Compiler				
	PGI	Sun	GCC	NAG	INTEL
mpipc	mpipc_pgi	mpipc_sun	mpipc_gcc	mpipc_nag	mpipc_intel
tornado	tornado_pgi	tornado_sun	tornado_gcc	tornado_nag	tornado_intel
squall	squall_pgi	squall_sun	squall_gcc	squall_nag	squall_intel

	MPI Setting			
	no MPI no OpenMP	no MPI yes OpenMP	yes MPI no OpenMP	yes MPI no OpenMP
blizzard	blizz_nMnO	blizz_nMyO	blizz_yMnO	blizz_yMyO

bb_make

```
#=====
# configure the Makefile and make the executables

./target_confmake.ksh ${BB_SLAVE}

# stop if not ok

check_error $? "bb_make: ./target_confmake.ksh ${BB_SLAVE}"

#=====
```

bb_run

- The script is divided into a “model loop” and a “post processing loop”
- The “model loop” looks for files starting with **exp.t** (like: exp.test_hat_jww), and starts it.
- The “post processing loop” looks for files starting with **post_** (like: **post_**hat_jww.bash), and starts it.
- If possible *qsub* and *llsubmit* are used

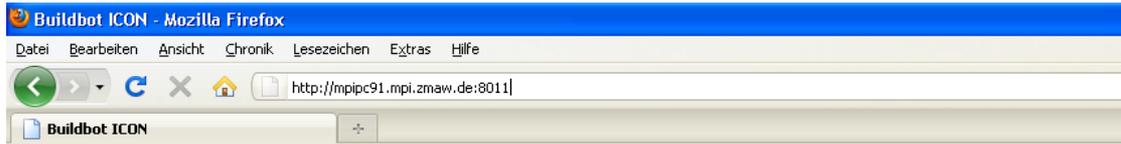
bb_run

```
if [ "$1" != "" ]      # --> from argument
then
    target="$1"
    source="argument"

elif [ "$BB_SLAVE" != "" ] then
    target="$BB_SLAVE"
    source='$BB_SLAVE'

else                    # --> as default
    target="default"
    source="$scriptname"
fi
```

Web Main Page Icon



Buildbot for ICON

Buildbot starts every night a test suite on different machines, using different compilers.

The test consists in:

1. removing files from the previous test
2. retrieving the most recent revision existing on "trunk/icon-dev"
3. running "configure&make" for a specified target machine and compiler
4. making test runs with the following sub steps:
 - i. make run scripts for the target machine and for all experiments, for which descriptor files named run/exp.test_* exist
 - ii. launch the test run scripts
 - iii. launch existing post processing scripts, named run/post_*

The success (in green) or failure (in red) of these tests is displayed on the following pages:

- [Waterfall display](#): Displays success/failure for all 4 steps listed above for all build targets, in chronological order.
- [Grid display](#): Displays overall success or failure for the 20 most recent revisions tested by Buildbot, including regular nightly builds and
- [Latest build](#): Displays the overall success or failure for the last test of each builder, ordered by builder.
- [List display](#): Displays the overall success or failure of the most recent builds, ordered by time of completion of the build tests.
- [Build slaves](#): Displays the status of all machines used by this Buildbot.

Postprocessing results are displayed here in comparison to a reference result from an earlier revision:

- [exp.test hat jww](#)

About Buildbot

- [Buildbot home page](#)
- [Buildbot version used](#)

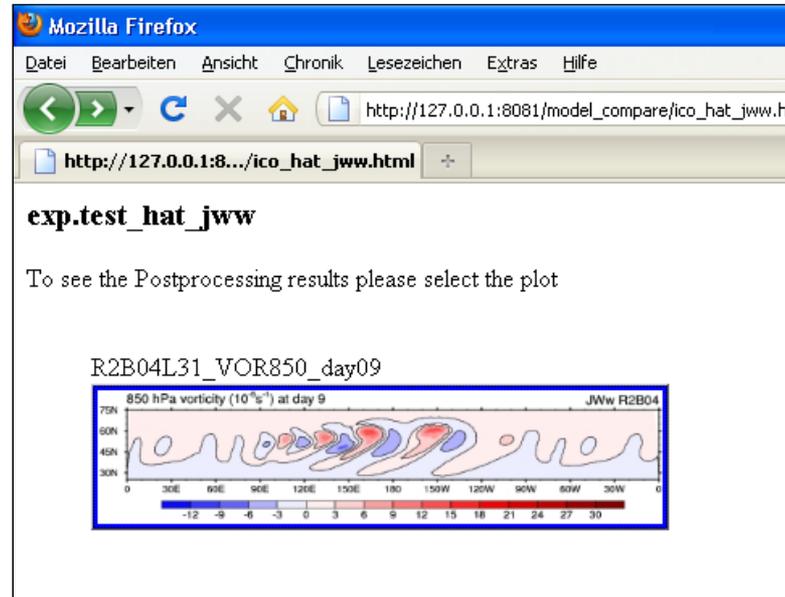
Model compare

- To have the possibility to compare the model output each run creates a graph (postprocessing loop).
- Each graph is collected and stored on the Server computer using a cronjob.
- The collection is done each hour.
- The cronjob creates a html-page to show each graph on one web-page.
- At the bottom of the web-page a link to the result of the day before can be found.
- In the future more experiment results can be included.

Model compare

Postprocessing results are displayed here in comparison to a reference result from an earlier revision:

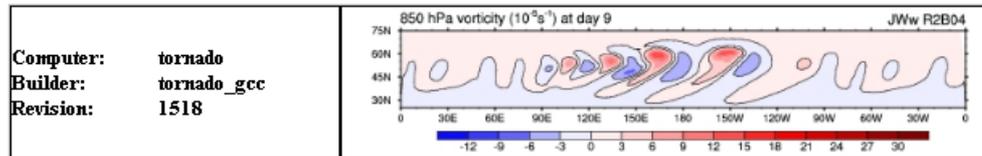
- [exp.test_hat_jww](#)



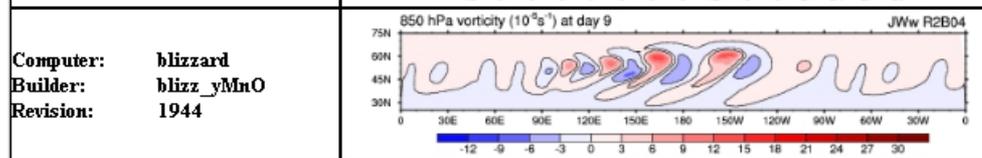
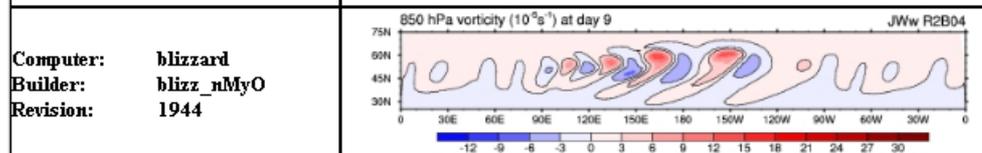
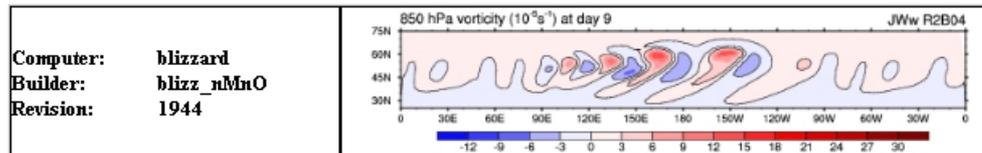
Model compare

13.07.2010

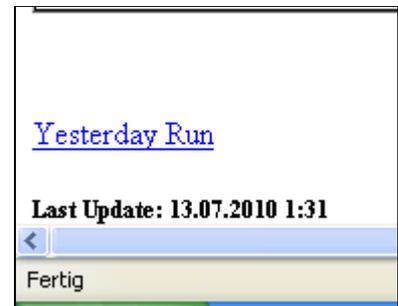
Reference



Daily Runs



Bottom of web-page



7. Problems

- The RETURN Status of a *llsubmit* call is always “OK”, but it can be not OK!
- If “*allowForce*” is set to “*False*” stopping of a running builder is not possible.
- If “*allowForce*” is set to “*True*” each user from the institute can trigger a builder.
- After a restart of the system a manual restart of client or server has to be done

Outlook

- Following BuildBot are running
 - **ECHAM** (mpipc91.mpi.zmaw.de:8010)
 - Compiler: GCC, NAG, INTEL, PGI, SUN
 - Systems: blizzard, tornado, squall, mpipc22
 - **ICON** (mpipc91.mpi.zmaw.de:8011)
 - Compiler: GCC, NAG, INTEL, PGI, SUN
 - Systems: blizzard, tornado, squall, mpipc22
 - **COSMOS** (mpipc91.mpi.zmaw.de:8012)
 - Models: asob, aso, as, ob, s
 - Systems: blizzard, tornado

Thank you



Max-Planck-Institut
für Meteorologie



MAX-PLANCK-GESELLSCHAFT



Universität Hamburg



International Max Planck Research School
on Earth System Modelling



KlimaCampus



FORSCHUNGSZENTRUM
in der HELMHOLTZ GEMEINSCHAFT



Max-Planck-Institut
für Meteorologie