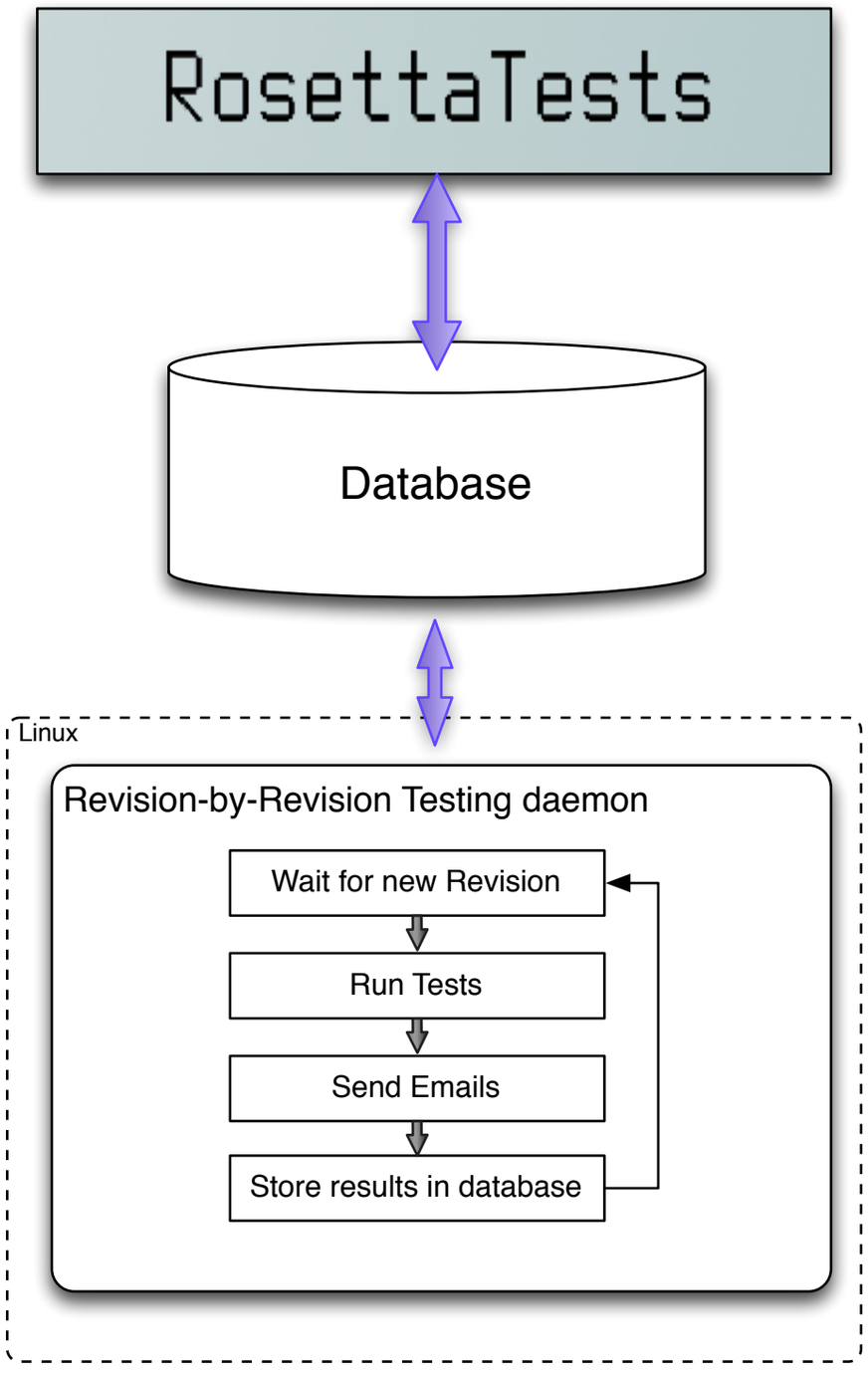
Benchmark 2

progress report and future directions

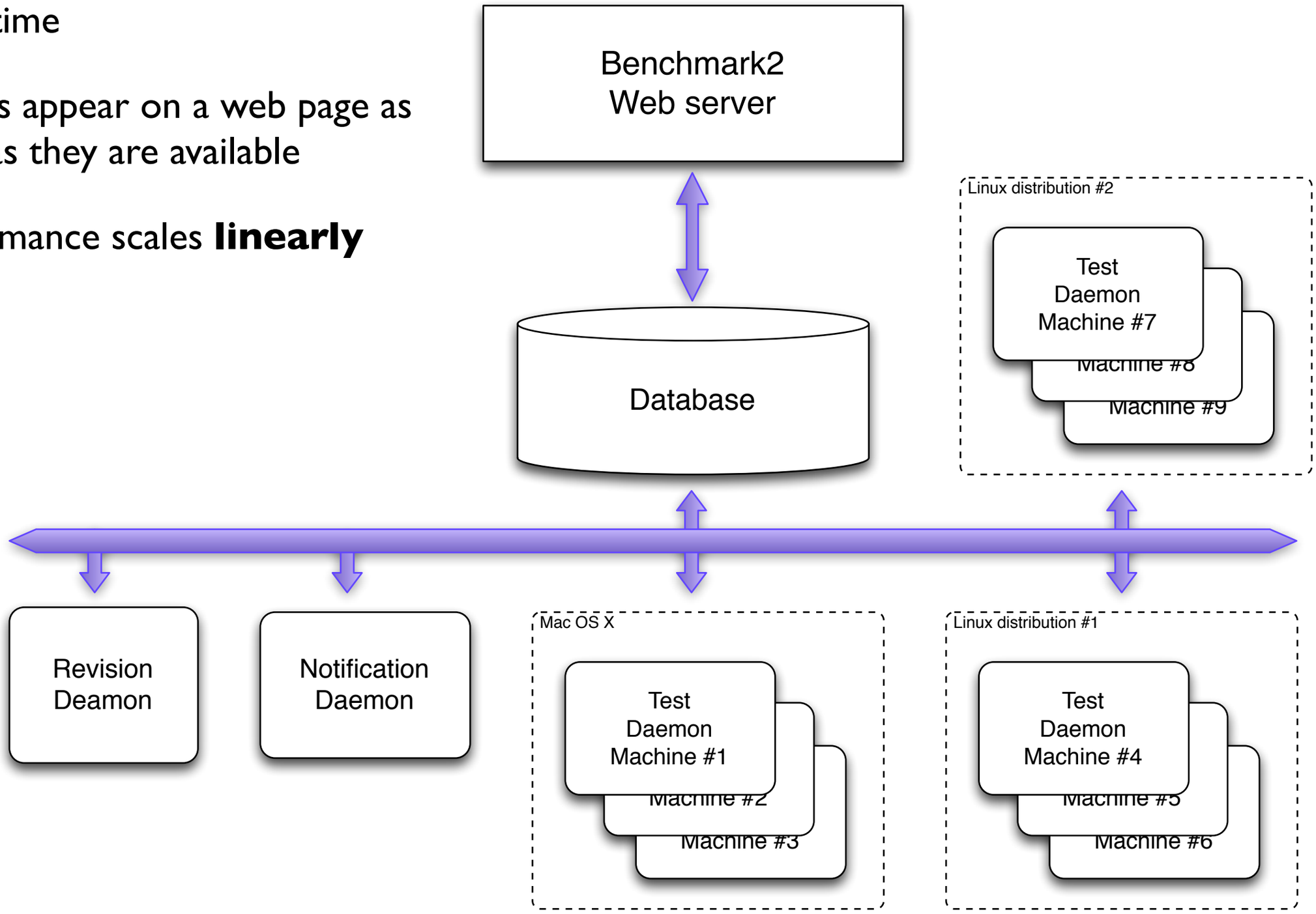
Sergey Lyskov GrayLab@JHU

Benchmark-1 (Current system):

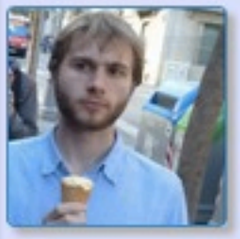


- Single platform
- Single machine
- Test only one revision at the same time
- Performance not scalable
- So we can run only a limited number of tests for each revision to keep testing time reasonable

- Run on Multiple platforms
- Test multiple revisions at the same time
- Results appear on a web page as soon as they are available
- Performance scales **linearly**



New website and GitHub integration



branch: master 「№57123」
Commited by: Justin R. Porter
GitHub commit link: 「28cd7d9ad8510ba6」
Difference from previous tested commit: code diff
Commit date: 2014-07-26 22:37:24

	linux.clang	linux.gcc	mac.clang
debug	■	■	■
release	■	■	■
unit	■	■	■

linux.PyRosetta | mac.PyRosetta | windows.PyRosetta
mysql | postgres | mpi | header only | library levels
static | integration.debug | integration | score

```
Contribute myriad bug fixes and additional unit testing to Broker 2.0 framework.
```

Summary | mac.PyRosetta | windows.PyRosetta | integration.debug 「4」 | integration 「5」

...

branch: master 「№57122」
Commited by: Rebecca Alford
GitHub commit link: 「aa8bac6640d404de」
Difference from previous tested commit: code diff
Commit date: 2014-07-26 20:40:16

	linux.clang	linux.gcc	mac.clang
debug	■	■	■
release	■	■	■
unit	■	■	■

linux.PyRosetta | mac.PyRosetta | windows.PyRosetta
mysql | postgres | mpi | header only | library levels
static | integration.debug | integration | score

```
Membrane Framework Updates: Part 2 in a series of membrane framework commits
================================================================================
Author: Rebecca Alford

Conformation:
  = Moved coordinate derived membrane info down into the membrane info object

Renaming:
  = Renamed InitialMembranePositionMover => MembranePositionFromTopologyMover
```

Benchmark-1 tests for each revisions:

linux.gcc.unit

linux.gcc.build.header

linux.gcc.build.levels

linux.gcc.build.PyRosetta

linux.gcc.build.debug

linux.gcc.build.release

linux.gcc.score

Benchmark-2 tests for each revisions:

linux.clang.build.debug

linux.clang.build.release

linux.clang.mysql.build.debug

linux.clang.postgres.build.debug

linux.clang.unit

linux.gcc.build.PyRosetta

linux.gcc.build.debug

linux.gcc.build.release

linux.gcc.integration.debug

linux.gcc.mpi.build.debug

linux.gcc.static.build.release

linux.gcc.unit

mac.clang.build.PyRosetta

mac.clang.build.debug

mac.clang.build.header

mac.clang.build.levels

mac.clang.build.release

mac.clang.integration

mac.clang.score

mac.clang.unit

windows.cl.build.PyRosetta

Benchmark-2 tests for each revisions:

linux.clang.build.debug

linux.clang.build.release

linux.clang.mysql.build.debug

linux.clang.postgres.build.debug

linux.clang.unit

linux.gcc.build.PyRosetta

linux.gcc.build.debug

linux.gcc.build.release

linux.gcc.integration.debug

linux.gcc.mpi.build.debug

linux.gcc.static.build.release

linux.gcc.unit

mac.clang.build.PyRosetta

mac.clang.build.debug

mac.clang.build.header

mac.clang.build.levels

mac.clang.build.release

mac.clang.integration

mac.clang.score

mac.clang.unit

windows.cl.build.PyRosetta

7→21

Tracking Custom Branches on Test Server:

- Testing server now track 5 branches
- We can customize which tests to run for each branch
- Could be *very* useful for team of developers who work on same project

- Julia Koehler and Rebecca Alford (rfalford12/mpframework_devel) wrote:

  Benchmark2 has been a really great resource for developing the membrane code. Frequently running all of the build, unit and integration tests helps us to keep our development compatible with master. Benchmark-2 is also a great collaboration tool - it helps Julia and I keep a running log of changes, and more easily merge our contributions.

Observers framework for Regression tests

- Add ‘observers’ text file to your integration tests dir
- Add lines with lines: `baranch1:email1 branch2:email2` ... for each branch you want to get notifications.
- Example:
  `master:sarel@weizmann.ac.il`
  `master:glapidoth@gmail.com`

- De-couple observer info specification from TestServer code
- Allow to specify multiple observers for multiple branches
- If score/integration tests got renamed or copied observers info is preserved

RosettaLog@gmail.com <rosettalog@gmail.com Jul 22 (1 day ago)
to sarel, glapidoth

Test(s) mac.clang.integration.splice_out, linux.gcc.integration.debug.splice_out got broken by following commit:

sarel commited 2b97188dd4adf678 [diff] to branch master:57095 with commit message:

```
adding two integration tests for splice_in and splice_out for non-
antibody cases
```

Demo

http://benchmark.graylab.jhu.edu/revisions

Future Directions

- Setup addition testing servers:
  - Linux
  - Mac
  - Windows (?)
- Cluster test, decouple HPC driver from testing scripts and port existing scientific and profile tests (as soon as our testing cluster upgraded to 64-bit OS)
- Implement binary search for best-effort tests
- Port performance benchmark and profile tests to Benchmark-2

Your feedback and suggestions are welcome!

Thank you!

Topics for discussion:

Topics for discussion:

- Do we need add more build tests? (MPI+Clang, iCC?)

Topics for discussion:

- Do we need add more build tests? (MPI+Clang, iCC?)
- Should we have additional status for Integration tests (ie “script failed”) that represent failure of executable to run instead of signaling regression tests changes?

Topics for discussion:

- Do we need add more build tests? (MPI+Clang, iCC?)
- Should we have additional status for Integration tests (ie “script failed”) that represent failure of executable to run instead of signaling regression tests changes?
- Metrics for Scientific Tests.

Topics for discussion:

- Do we need add more build tests? (MPI+Clang, iCC?)
- Should we have additional status for Integration tests (ie “script failed”) that represent failure of executable to run instead of signaling regression tests changes?
- Metrics for Scientific Tests.
  - What kind of reports tools do we want? (simple plots? something more powerful?)

Topics for discussion:

- Do we need add more build tests? (MPI+Clang, iCC?)
- Should we have additional status for Integration tests (ie “script failed”) that represent failure of executable to run instead of signaling regression tests changes?
- Metrics for Scientific Tests.
  - What kind of reports tools do we want? (simple plots? something more powerful?)
  - Can we manage to have scientific tests results to be non-incremental? (its not clear how to manage scheduling for best-effort tests which require comparison with previous runs to deliver results)

http://benchmark.graylab.jhu.edu/submit

Submit Custom Revision for Testing

65520983418baf5 ▶

Commited by: 「Labonte」
GitHub commit link: 「65520983418baf54c31541a61837c364d777345f」
Commit date: 2014-07-22T21:31:04Z

```
ResdidueProperties: improvements to auto-code-generation script

update_residue_properties.py now...
...only re-writes the files if changes have occured,
...ignores duplicated properties, and
...is more modular.

----------------------------------------
Unit test status: Not Run
Integration test status: Not Run
```

Rosetta Build Tests

- ☐ linux.clang.build.debug
- ☐ linux.clang.build.release
- ☐ linux.gcc.build.debug
- ☐ linux.gcc.build.release
- ☐ mac.clang.build.debug
- ☐ mac.clang.build.release
- ☐ static

none all

Code Integrety Tests

- ☐ header only
- ☐ library levels

none all

Unit Tests

- ☐ linux.clang.unit
- ☐ linux.gcc.unit
- ☐ mac.clang.unit

none all

Regression Tests

- ☐ integration
- ☐ score

none all

PyRosetta Tests

- ☐ linux.PyRosetta
- ☐ mac.PyRosetta
- ☐ windows.PyRosetta

none all

Other Tests

- ☐ mpi
- ☐ mysql
- ☐ postgres

none all

Now we can comment on revision:

branch: _commits_ 「№10」
Commited by: Labonte
GitHub commit link: 「912b2aef1a6c1606」

Commit date: 2014-07-17 01:53:29

	linux.clang	linux.gcc	mac.clang
debug			
release			
unit			

mac.PyRosetta | mac.gcc.build.PyRosetta | windows.PyRos

mpi | header only | library levels | static | integration | scc

How the hell did that entire file get wiped?

Labonte 1 week
Sergey, I don't think the new integration tests are compared to the ref from when I branched from master.

Rosetta Benchmark 0 seconds
They compared to latest successful run from the master (i.e. run where build did not fail)

Leave a comment...

- So we can 'add' info to accidental commit messages
- And we can comment on test results and such