LAMMPS Software Development on GitHub

Richard Berger
Anders Hafreager

LAMMPS Workshop 2017

August 1, 2017
LAMMPS GitHub Tutorial

Target Audience

▶ LAMMPS Developers
▶ Package Maintainers
▶ Contributors

Agenda

1. GitHub Workflow Overview
2. Git Introduction
3. Hands-on
Outline

Background

Releases

Issue Tracking

Pull Requests
Some History

LAMMPS started as Fortran; rewritten in C++

- make it easy to add new kernels (“styles”)
- implemented as derived polymorph classes
- top-level code is an instance of a composite of instances of selected styles/classes

Linear development

- originally not using any dedicated source code management tools
- later adapted Subversion for version control

Continuous development model

- changes released quickly and frequently
- core code was supposed to always work
LAMMPS in numbers

- 750+ styles (1250+ if you count accelerator variants)
- 31 packages
- 30 USER packages
- 21 external libraries
- One dozen features enabled through compiler flags (e.g. `-DLAMMPS_PNG`, `-DLAMMPS_FFMPEG`)
Commit History in the old workflow

<table>
<thead>
<tr>
<th>Release Tag</th>
<th>Date</th>
<th>Commits</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>r15407</td>
<td>305</td>
<td>2016-07-30</td>
<td>sjplimp (289), athomps (11), stamoor (5)</td>
</tr>
<tr>
<td>r15061</td>
<td>369</td>
<td>2016-05-17</td>
<td>sjplimp (329), stamoor (26), athomps (14)</td>
</tr>
<tr>
<td>r14624</td>
<td>261</td>
<td>2016-02-15</td>
<td>sjplimp (222), stamoor (24), athomps (15)</td>
</tr>
<tr>
<td>r14304</td>
<td>368</td>
<td>2015-12-08</td>
<td>sjplimp (286), athomps (71), stamoor (11)</td>
</tr>
<tr>
<td>r13864</td>
<td>282</td>
<td>2015-08-10</td>
<td>sjplimp (245), athomps (25), stamoor (12)</td>
</tr>
<tr>
<td>r13475</td>
<td>319</td>
<td>2015-05-15</td>
<td>sjplimp (290), athomps (14), pscrozi (8), stamoor (7)</td>
</tr>
</tbody>
</table>

- Attributions would go into README files and lammps.sandia.gov/authors.html
- Contributions and Integration work not visible
LAMMPS Development Pyramid

“the big boss”
Steve Plimpton

core developers
2x @Sandia, 2x @Temple
core functionality, maintenance, integration

package maintainers
> 30, mostly user pkgs, some core

single/few style contributors
> 100, user-misc and others

Feedback from mailinglist, GitHub Issues
Why change the workflow?

continuing growth increases maintainance effort

accelerated styles increase code complexity
base class changes may break derived classes

disruptive changes to core may need contributions from multiple developers
need a facility for **concurrent** development and open/reviewable communication between contributors/maintainers

offload some code integration to contributor
provide feedback on merging effort
Toolchain

**git**
- distributed version control system
- efficient handling of feature branches
- powerful merging tool

**GitHub**
- public and well performing git hosting
- also communication platform: discussions tied to issues or pull requests
- provides API and triggers for testing tasks

**Jenkins**
- continuous integration and testing platform
- allows us to define compilation and testing pipelines
- operates on multiple platforms
Workflow Documentation

- **LAMMPS Contributing Guidelines**
  https://github.com/lammps/lammps/blob/master/.github/CONTRIBUTING.md

- **LAMMPS GitHub Tutorial**

Note
These are best practices which developed over time. They're not set in stone and we continue to improve it.
LAMMPS on GitHub

- [http://github.com/lammps/lammps](http://github.com/lammps/lammps)
  - public development repository
  - issue tracking
  - changes from Sandia SVN are integrated here before new releases
  - contributions are processed as Pull Requests for code review and testing
Release Policy

Patch Releases
- a collection of bugfixes and new features
- every few weeks
- posted on website as tarball

Stable Releases
- a release marked as stable after longer periods of testing
- derived from latest patch release
- period of feature-freeze and only accepting bugfixes
- released about every 3-4 months
- posted on website as tarball
Public Development on GitHub

**master branch**
- main development branch
- the latest and greatest development version of LAMMPS
- while we try really hard to keep it stable, things might break for brief periods of time
- even changes from Sandia are integrated back by Steve to GitHub via Pull Requests

**unstable branch**
- a bit older than **master**
- follows patch releases

**stable branch**
- a bit older than **unstable**
- follows stable releases
LAMMPS Branches

- **master**
  - S

- **unstable**
  - S

- **stable**
  - S

- **change**
  - change

- **patch release**
  - patch release

- **stable release**
  - stable release
LAMMPS Branches

- master
- unstable
- stable

- change
- patch release
- stable release
LAMMPS Branches

- master
- unstable
- stable

current version (HEAD of branch)

- change
- patch release
- stable release
LAMMPS Branches

- **master**
- **unstable**
- **stable**

**current version (HEAD of branch)**

- **change**
- **patch release**
- **stable release**
LAMMPS Branches

master

unstable

stable

change

patch release

stable release
LAMMPS Branches

- master
  - S
  - P

- unstable
  - S
  - P

- stable
  - S

change

patch release

stable release
LAMMPS Branches

- **master**: S P P
- **unstable**: S P P
- **stable**: S

- **change**: ○
- **patch release**: P
- **stable release**: S
LAMMPS Branches

- **master**
  - S
  - P
  - P

- **unstable**
  - S
  - P
  - P
  - P

- **stable**
  - S

- **change**
- **patch release**
- **stable release**
LAMMPS Branches

- **master**
  - S
  - P
  - S

- **unstable**
  - S
  - P
  - S

- **stable**
  - S
  - P
  - S

- **change**
- **patch release**
- **stable release**
LAMMPS Branches

- **master**
  - S
  - P
  - S

- **unstable**
  - S
  - P
  - S

- **stable**
  - S
  - P
  - S

- ![change](change.png)
- ![patch release](patch_release.png)
- ![stable release](stable_release.png)
LAMMPS Branches

master

unstable

stable

change

patch release

stable release
LAMMPS Branches

- **master**
  - S
  - P
  - S
  - P

- **unstable**
  - S
  - P
  - S
  - P

- **stable**
  - S
  - P
  - S

- **change**
- **patch release**
- **stable release**
LAMMPS Branches

- master
- unstable
- stable

○ change
P patch release
S stable release
LAMMPS stable release 31 Mar 2017

akohlmey released this on Mar 30 · 659 commits to master since this release

Notable new features in this release are:

- refactoring of the neighbor list construction to be more modular and extensible from packages. active neighbor list modules are reported in the output
- updates, performance improvements and new styles for the KOKKOS and USER-INTEL packages
- bugfixes and refactoring of parts of AIREDO and AIREDO-M addressing some of the occasionally observed instabilities with the LAMMPS implementation
Releases on GitHub

Downloads

- Source code (zip)
- Source code (tar.gz)

patch_31Mar2017

hash: 56b9a

Download [zip](#) [tar.gz](#)
Releases on GitHub

- patch_24Jul2017
  - 6 days ago
  - d59b7e4
  - zip
  - tar.gz

- patch_6Jul2017
  - 24 days ago
  - 4339779
  - zip
  - tar.gz

- patch_23Jun2017
  - on Jun 23
  - 1370383
  - zip
  - tar.gz

- patch_20Jun2017
  - on Jun 20
  - 87c028e
  - zip
  - tar.gz

- patch_19May2017
  - on May 19
  - 2225f8c
  - zip
  - tar.gz

- patch_4May2017
  - on May 4
Issue tracking

Usage:

- tracking bugs
- feature requests
- planning new features

Labels

- applied by maintainers
- used to group by topic

Assignment

- One or more contributors
- Meaning: ownership and responsible for resolution
Implement short neighborlists on manybody potentials in GPU package #543

andeplane opened this issue on Jun 22 - 3 comments

andeplane commented on Jun 22

In CPU and KOKKOS versions of vashishta, sw and tersoff, we use short neighbor lists that only contains neighbors within cutoff. This is especially useful in vashishta where 3-body forces usually have much smaller cutoff.

The GPU package is nice since it supports single/mixed precision and works in Atomify, so I'm interested in porting this feature into the GPU package as well. I'm not so familiar with how it actually does neighbor list building etc, so if anyone knows how to do this easily, feel free to do it :D If not I'll probably try to understand how it's done.

stanmoore1 added bugfix enhancement gpu_package and removed bugfix labels on Jun 26

andeplane commented 26 days ago

I have been in contact with @ndtrung81 and Mike on email about this. I think I know how to do it and will get help on the way. Probably done within a month or two.

stanmoore1 commented 19 days ago

@andeplane, glad to hear
Writing issue descriptions and comments: Markdown Syntax

Headers

```
# Header 1
## Header 2
### Header 3
```

Ordered Lists

```
1. first
2. second
3. third
```

Code Snippets

```
``language
code
```

Unordered Lists

```
* first
* second
* third
```

Links

[Link](http://www.google.com)
Writing issue descriptions and comments: Markdown Syntax

Task list with check boxes

* [ ] Task A
* [x] Task B
* [ ] Task C

Images and Attachments

- Each comment can add images or files (certain types). Simply drag & drop them into the editor.

Mention other users

By adding @username to your message, you create a reference to that user. They will also get notified of that comment. This is a way to direct the conversation in an issue or pull request.
Not An Issue

- Issues are for code development related topics only!
- **Do not create an issue to ask how to use LAMMPS or discuss physics!**
- They will be ignored and closed. ⇒ Use the mailing list instead.
Working on Code

- So you’ve assigned yourself to an issue and working on a fix
- Or you are implementing a new feature
- What now?
Working on Code

- So you’ve assigned yourself to an issue and working on a fix
- Or you are implementing a new feature
- What now?

GitHub Workflow

1. Get the latest code (master)
2. Create a git branch to work on
3. Save your changes in that branch
4. Create a pull request
5. Follow core developer’s instructions and modify your contribution accordingly
6. Wait for merge
GitHub Workflow

Create feature branch

master branch
GitHub Workflow

Create **feature** branch

apply changes

*master* branch
GitHub Workflow

Create **feature** branch

apply changes
GitHub Workflow

Create **feature** branch

*master* branch

apply changes

Submit Pull Request
GitHub Workflow

Create feature branch

master branch

Apply changes

Submit Pull Request
GitHub Workflow

Create **feature** branch

apply changes

**master** branch

discuss & test changes

Submit Pull Request
GitHub Workflow

Create feature branch

Apply changes

Discuss & test changes

Submit Pull Request
GitHub Workflow

master branch

Create feature branch

apply changes

discuss & test changes

Submit Pull Request
GitHub Workflow

Create **feature** branch

apply changes

Discuss & test changes

Submit Pull Request

Merge with master
Pull Requests: Contents

- a branch containing a sequence of changes
- summarized as the list of commits and files changed
- the goal is to combine this sequence of changes to the master branch
- like an issue, a PR can have a rich-text description
- For your convenience, any new pull request will start with a template text which you are supposed to fill out and edit accordingly
Pull Requests: Automatic Testing

- Once a PR is submitted, your changes will be automatically tested
- Pushing further changes to your branch will be appended to the PR and be tested again
- Click “Details” to get to Jenkins
Build #928 (Jul 28, 2017 4:11:30 PM)

- No changes.
- GitHub PR #600, Commit changed

This run spent:
- 8 sec waiting in the queue;
- 5 min 54 sec building on an executor;
- 6 min 2 sec total from scheduled to completion.

GNU Make + GNU C Compiler Warnings: 188 warnings
- 1 fixed warning

Static Analysis Warnings: 183 warnings
- 1 fixed warning
Pull Requests: Automatic Testing

git push → GitHub

notifies → Jenkins

Compilation

Testing

report result

report result
Pull Requests: Automatic Testing

All checks have passed
4 successful checks

- lammps/pull-requests/build-docs-pr — head run ended
- lammps/pull-requests/openmpi-pr — head run ended
- lammps/pull-requests/serial-pr — head run ended
- lammps/pull-requests/shlib-pr — head run ended

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request or view command line instructions.
GitHub Workflow - Multiple Feature Branches

Branches can have different starting points. As long as they have a common point in time, git will try to figure out how to merge these two sequences.
Feature branches

- branches allow you to work on more than one topic/feature
- changes in feature A unrelated to changes in feature B ⇒ use separate branches
- the time it takes to finish one topic/feature may differ from another
- by separating them you can more easily merge completed work, without having to complete all the other features you are working on
- use short descriptive names for branches (e.g. doc_corrections, fix_segmentation_fault, add_pair_style_lj)
- remember, you can write a more detailed description in the Pull Request
Real World Examples: AIREBO bugfix

https://github.com/lammps/lammps/issues/59

- Discussions on LAMMPS Mailinglist / Emails to Steve/Axel
- Multiple independent groups were having issues, some comparing with their own code
- Issue created on May 12, 2016
- We fixed some smaller bugs, but we knew it wasn’t the root cause
- A small group of GitHub users assembled, sharing thoughts, code and examples for testing
- Pull requests followed in the course of the issue discussions
- One month ago we could finally close this issue
Real World Examples: MEAM C++ implementation

https://github.com/lammps/lammps/issues/174

- Issue created on Sep 9, 2016
- A low priority TODO item
- Months later, someone picked it up because they needed it in a new pair style
- A few weeks later we had a complete rewrite
Commit History in the old workflow

<table>
<thead>
<tr>
<th>Release Tag</th>
<th>Date</th>
<th>Commits</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>r15407</td>
<td>305</td>
<td>2016-07-30</td>
<td>sjplimp (289), athomps (11), stamoor (5)</td>
</tr>
<tr>
<td>r15061</td>
<td>369</td>
<td>2016-05-17</td>
<td>sjplimp (329), stamoor (26), athomps (14)</td>
</tr>
<tr>
<td>r14624</td>
<td>261</td>
<td>2016-02-15</td>
<td>sjplimp (222), stamoor (24), athomps (15)</td>
</tr>
<tr>
<td>r14304</td>
<td>368</td>
<td>2015-12-08</td>
<td>sjplimp (286), athomps (71), stamoor (11)</td>
</tr>
<tr>
<td>r13864</td>
<td>282</td>
<td>2015-08-10</td>
<td>sjplimp (245), athomps (25), stamoor (12)</td>
</tr>
<tr>
<td>r13475</td>
<td>319</td>
<td>2015-05-15</td>
<td>sjplimp (290), athomps (14), pscrozi (8), stamoor (7)</td>
</tr>
</tbody>
</table>

- Attributions would go into README files and lammps.sandia.gov/authors.html
- Contributions and Integration work not visible
## Commit History in the new workflow

<table>
<thead>
<tr>
<th>Release Tag</th>
<th>Commits</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>patch_24Jul2017</td>
<td>144</td>
<td>Axel Kohlmeyer (74), sjplimp (21), Abdoreza Ershadinia (15), Markus Hoehnerbach (8), Abdo (6), Ryan S. Elliott (5), Giacomo Fiorin (5), Max Veit (4), Steve Plimpton (3), Christoph Junghans (2), H. Metin Aktulga (1)</td>
</tr>
<tr>
<td>patch_6Jul2017</td>
<td>93</td>
<td>Axel Kohlmeyer (47), Sebastian Hütter (18), sjplimp (12), Stefan Paquay (6), Steve Plimpton (5), Andrew Jewett (3), Stan Moore (2)</td>
</tr>
<tr>
<td>patch_23Jun2017</td>
<td>28</td>
<td>Axel Kohlmeyer (18), sjplimp (5), Steve Plimpton (5)</td>
</tr>
<tr>
<td>patch_20Jun2017</td>
<td>190</td>
<td>Axel Kohlmeyer (105), sjplimp (30), Emile Maras (10), Anders Hafreager (10), Lars Pastewka (8), Richard Berger (7), Steve Plimpton (5), Stefan Paquay (5), Stan Moore (4), dstelter92 (3), Oliver Henrich (2), Markus Hoehnerbach (1)</td>
</tr>
<tr>
<td>patch_19May2017</td>
<td>132</td>
<td>Axel Kohlmeyer (62), DallasTrinkle (22), sjplimp (21), Richard Berger (19), Steve Plimpton (7), Giacomo Fiorin (1)</td>
</tr>
<tr>
<td>patch_4May2017</td>
<td>39</td>
<td>Axel Kohlmeyer (25), sjplimp (5), Stan Moore (4), ketankhare (3), Steve Plimpton (2)</td>
</tr>
</tbody>
</table>
Questions?