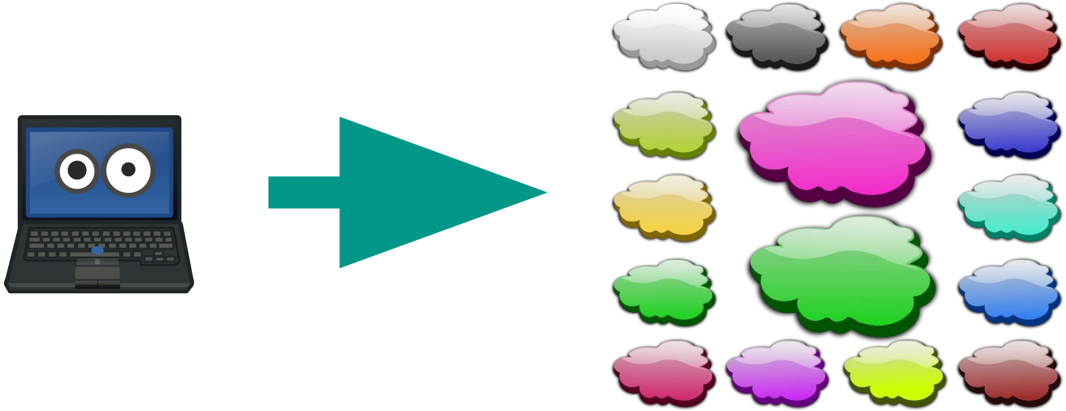
Automatic Dependency Management for Scientific Applications on Clusters

Ben Tovar*, Nicholas Hazekamp,
Nathaniel Kremer-Herman,
Douglas Thain

UNIVERSITY OF
NOTRE DAME

Where users are



Scientist says: ”This demo task runs on my laptop, but the real application has thousands of tasks. I wonder if it can be run on this cloud/grid/opportunistic resources I have heard so much about?”

Where we come in

- **Sure! What do you need?**
- Linux.
- **Which version?**
- …
- **Let’s say RHEL7 for starters. I guess you need python?**
- Yes!
- **Which version?**
- …
- **Let’s say python 2.7. Anything else?**
- No. I don’t think so.

Where we end up

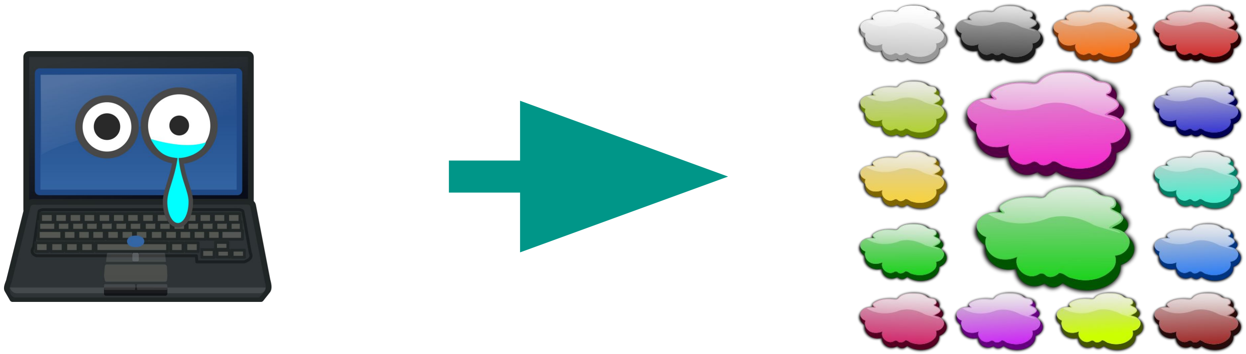
- Hey, it doesn’t work. The error says that `blastn` is missing.

Could you install it in all the sites and machines I may end up using?

Where we end up

- Hey, it doesn’t work. The error says that `blastn` is missing.

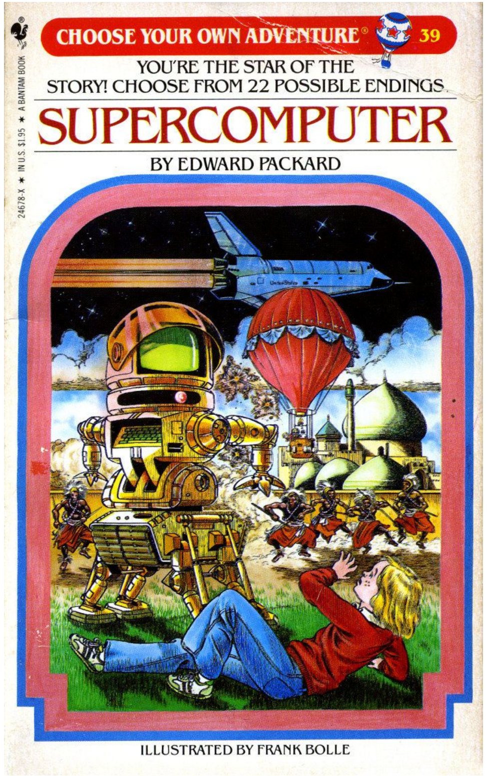
Could you install it in all the sites and machines I may end up using?



Three problems

1. Finding dependencies is an interactive discovery adventure for the user

We may point the user in some direction, but it is not possible to know all the dependencies for all the software a scientist/postdoc/grad-student may end up using.


CHOOSE YOUR OWN ADVENTURE® 39
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 22 POSSIBLE ENDINGS.
SUPERCOMPUTER
BY EDWARD PACKARD
ILLUSTRATED BY FRANK BOLLE
24678-X ★ IN U.S. $1.95 ★ A BANTAM BOOK

Three problems

2. The sysadmins don't have the time to install all of the software

Even if they had the time, they couldn't anyway in opportunistic resources.

And even if they could, scientific software has many ad-hoc custom installations.

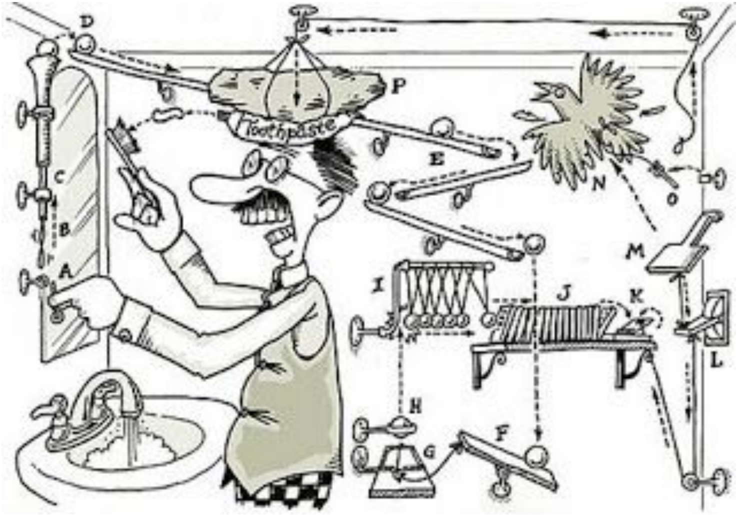
And even if it didn't, it changes very rapidly.

I'M TIRED GERALD.
I'M TIRED OF EVERYONE ALWAYS ASKING ME TO FIX THEIR PROBLEMS.

Three problems

3. Not all dependencies are software packages.

Operating systems, filesystem mounts, network access, among others.



I know what you are thinking

I know what you are thinking

Let’s use containers!

(or vms if you don’t think yourself as a whippersnapper)

’ problem is...

Containers start like this:

We need to fill the container by solving the aforementioned problems.

Containers can be used later to deliver the solutions we may find.

Our solution

The **vc3-builder**, a command-line tool for deploying software environments on clusters.

```
vc3-builder
   --require-os redhat:7
   --mount /scratch=/data
   --require python:2.7 -- myapp ...my args...
```

Use case

```
vc3-builder --require python:2.7.5 --require ncbi-blast myapp
```

Use case

```
vc3-builder --require python:2.7.5 --require ncbi-blast myapp
```

compute node

with:

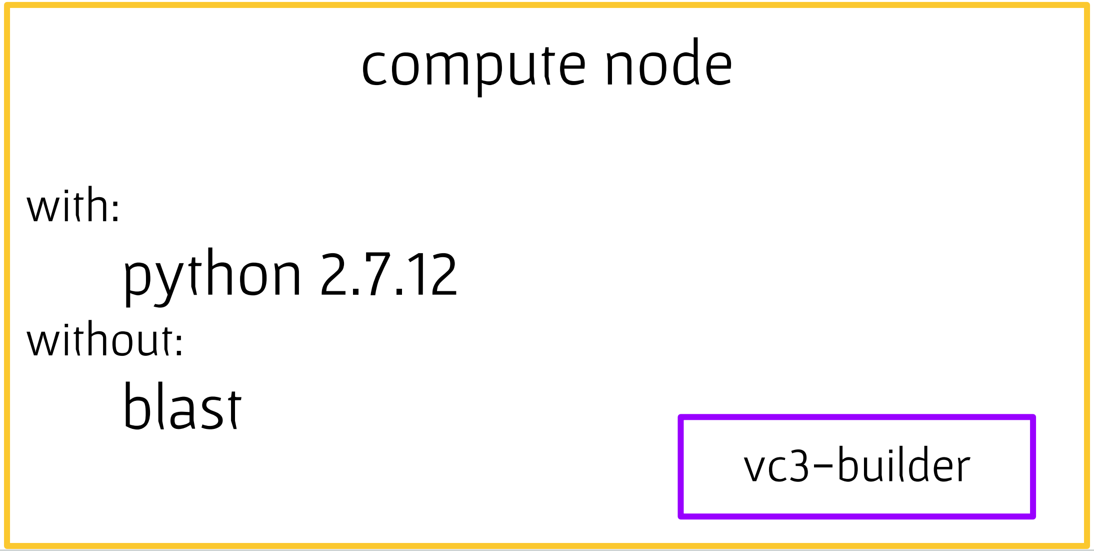
python 2.7.12

without:

blast

Use case

```
vc3-builder --require python:2.7.5 --require ncbi-blast myapp
```



compute node

with:

python 2.7.12

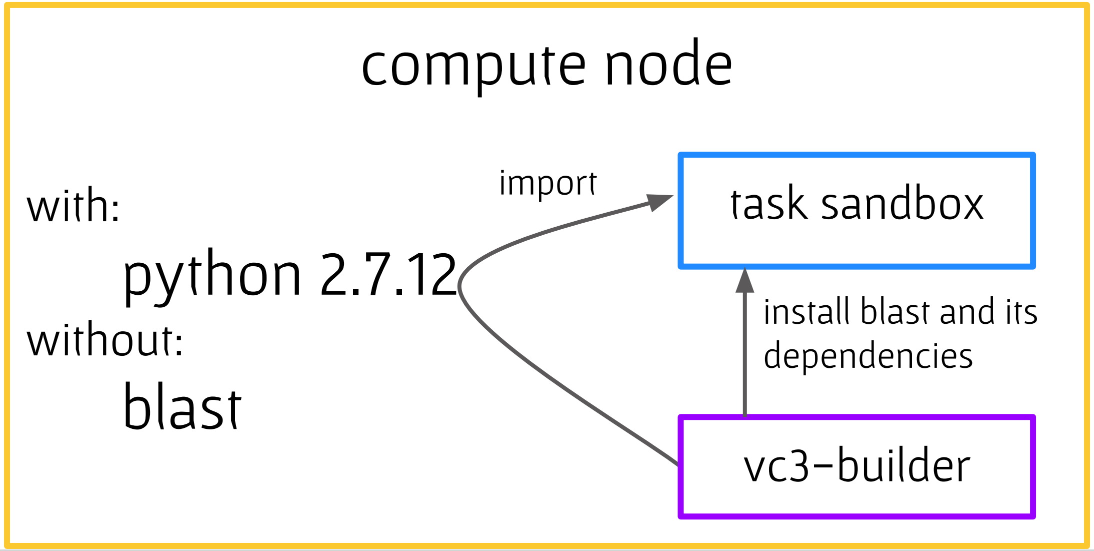
without:

blast

vc3-builder

Use case

```
vc3-builder --require python:2.7.5 --require ncbi-blast myapp
```



compute node

with:

python 2.7.12

without:

blast

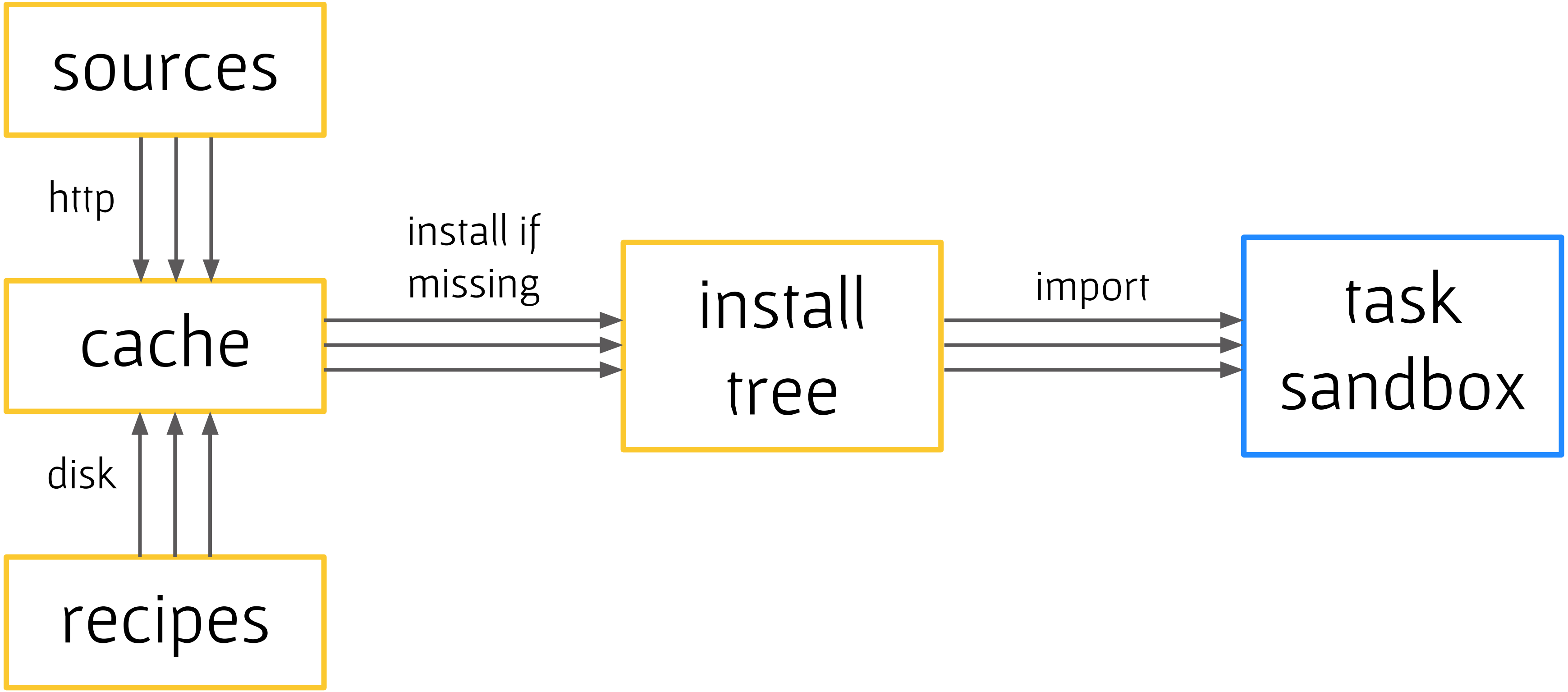
import

task sandbox

install blast and its dependencies

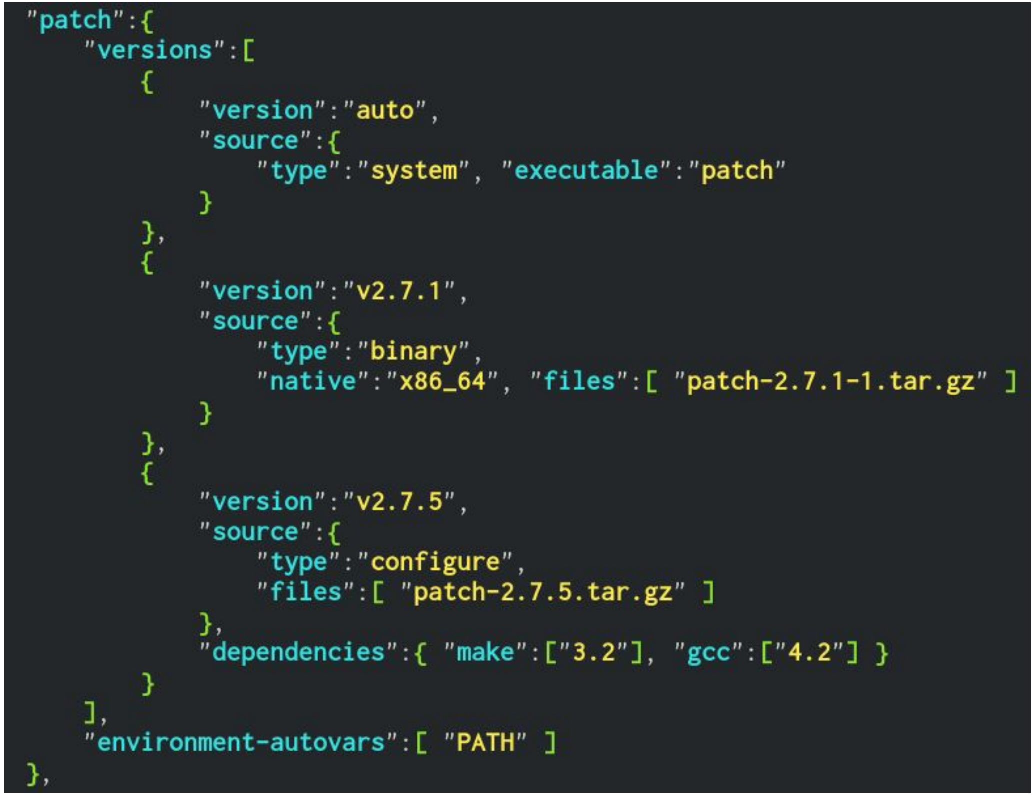
vc3-builder

Architecture



Help users with dependency discovery

1. As users locally discover dependencies using a clean sandbox, they write shell (sh) recipes in a JSON document.



```
"patch":{
    "versions":[
        {
            "version":"auto",
            "source":{
                "type":"system", "executable":"patch"
            }
        },
        {
            "version":"v2.7.1",
            "source":{
                "type":"binary",
                "native":"x86_64", "files":[ "patch-2.7.1-1.tar.gz" ]
            }
        },
        {
            "version":"v2.7.5",
            "source":{
                "type":"configure",
                "files":[ "patch-2.7.5.tar.gz" ]
            },
            "dependencies":{ "make":["3.2"], "gcc":["4.2"] }
        }
    ],
    "environment-autovars":[ "PATH" ]
},
```

Rely less on sysadmins for installation

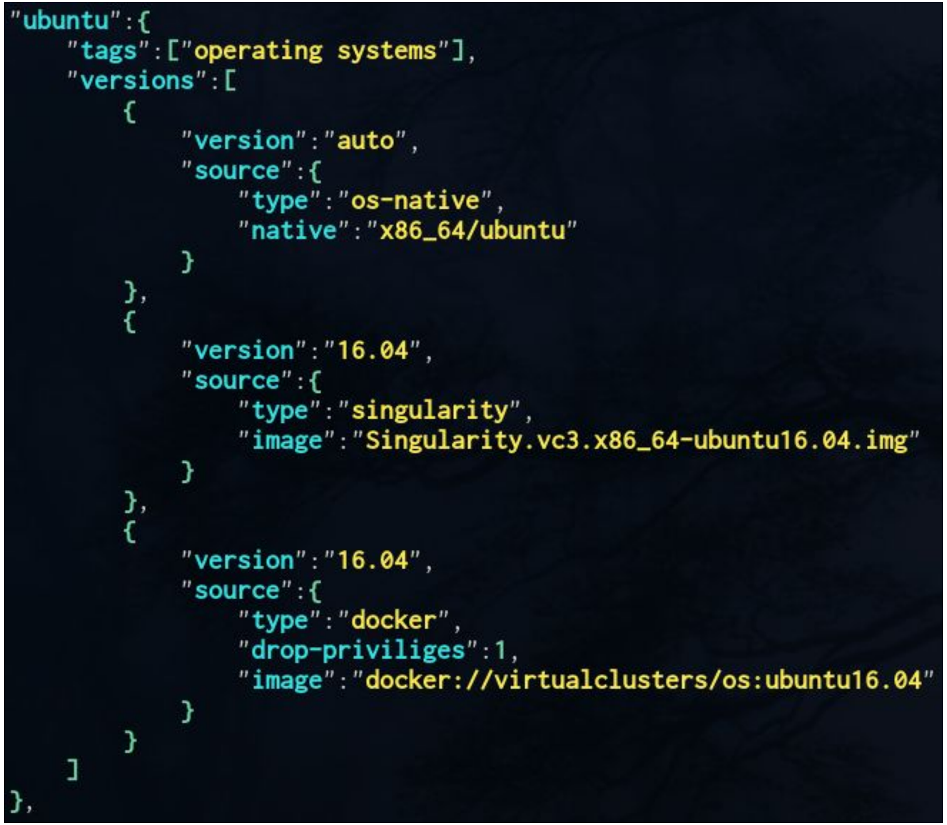
2. All vc3-builder actions are done only with the privileges the user already has (e.g., no sudos).

- Everything is installed to a directory the user has write access (e.g., $HOME/myOwnDir)

- As much as possible, dependencies already available in the host system are not re-installed.

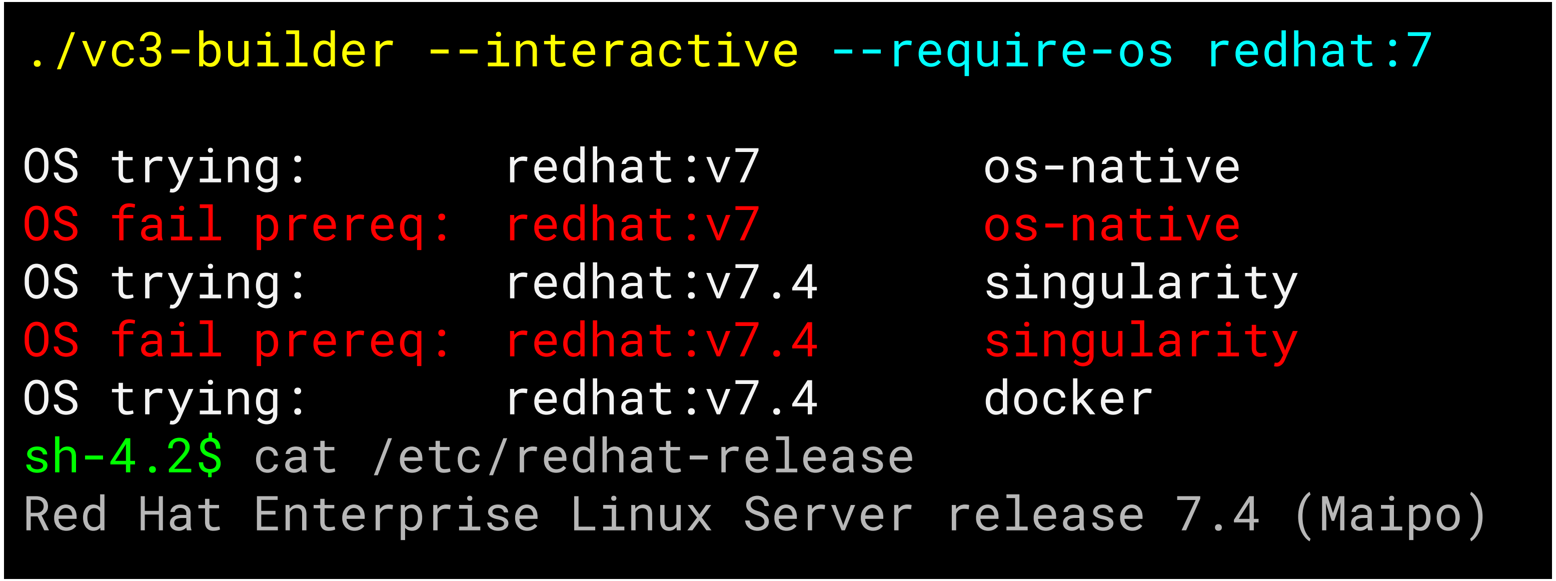
More than software dependencies

3. Allow for the dependencies on particular operating systems, filesystems, or mount points to be described.



```
"ubuntu":{
    "tags":["operating systems"],
    "versions":[
        {
            "version":"auto",
            "source":{
                "type":"os-native",
                "native":"x86_64/ubuntu"
            }
        },
        {
            "version":"16.04",
            "source":{
                "type":"singularity",
                "image":"Singularity.vc3.x86_64-ubuntu16.04.img"
            }
        },
        {
            "version":"16.04",
            "source":{
                "type":"docker",
                "drop-priviliges":1,
                "image":"docker://virtualclusters/os:ubuntu16.04"
            }
        }
    ]
},
```

Technology independence



```
./vc3-builder --interactive --require-os redhat:7

OS trying:       redhat:v7        os-native
OS fail prereq:  redhat:v7        os-native
OS trying:       redhat:v7.4      singularity
OS fail prereq:  redhat:v7.4      singularity
OS trying:       redhat:v7.4      docker
sh-4.2$ cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.4 (Maipo)
```

These were not popular a few years ago, and new technologies will be a few years hence.

The vc3-builder does not tie down the user to a single technology.

An important consideration

Lightweight bootstrap

vc3-builder is a completely self-contained program, including recipes.

Meant to be easy to deploy.

It can be compiled to a truly statically linked binary.

```
wget github.com/vc3-project/vc3-builder/releases/download/release/v0.1.0/vc3-builder
chmod 755 vc3builder
./vc3-builder --require …
```

Other considerations

Batch context

Interaction with the user is not desirable.

Installation as a preliminary step to job execution.

Dynamic composition

Use what's already there.

Each package to its own sandbox.

The builder can be used inside another builder instance.

Differences with established players

	spack	*nix	vc3-builder
Lightweight bootstrap	tarball + env script, 104 MB.	tarball + env script, 67 MB	single executable, 600 KB
Batch context	meant to run in headnode	meant to run in headnode	part of job execution
OS and filesystem dependencies	no	no	wrapping singularity, docker and the parrot virtual file system
Focus	High Performance Computing (HPC)	reproducibility	High Throughput Computing (HTC)
Bit-to-bit guarantees	system packages discouraged	yes, by compiling everything from source (even the compiler), binaries if write access to /	system packages encouraged
Recipes	python abstractions	nix expression language	sh commands as JSON strings
Number of recipes	thousands	thousands	dozens
Dependency resolution	powerful	powerful	primitive

What do we gain?

lightweight bootstrap

+ batch context

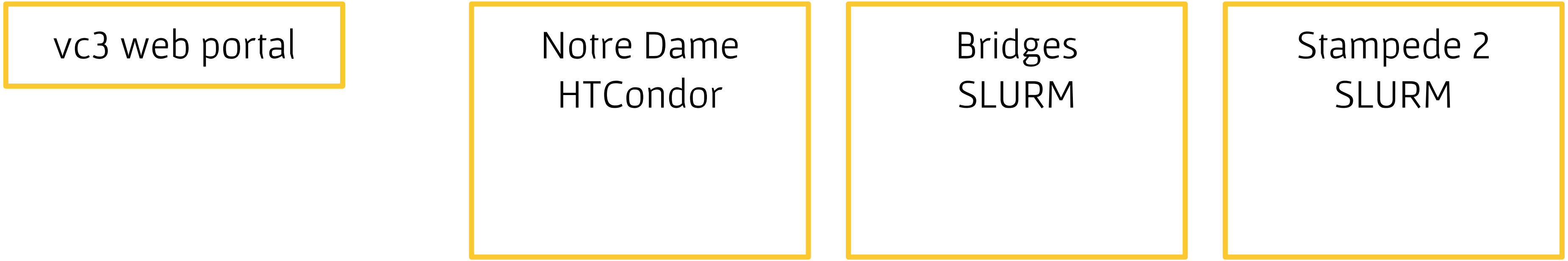
---

harness batch/cloud resources

What do we gain?

The vc3-builder was written in the context of the **V**irtual **C**lusters for **C**ommunity **C**omputation (VC3) project.

Users go to a **website** and create **short lived clusters** across **heterogeneous resources.**



vc3 web portal

Notre Dame HTCondor

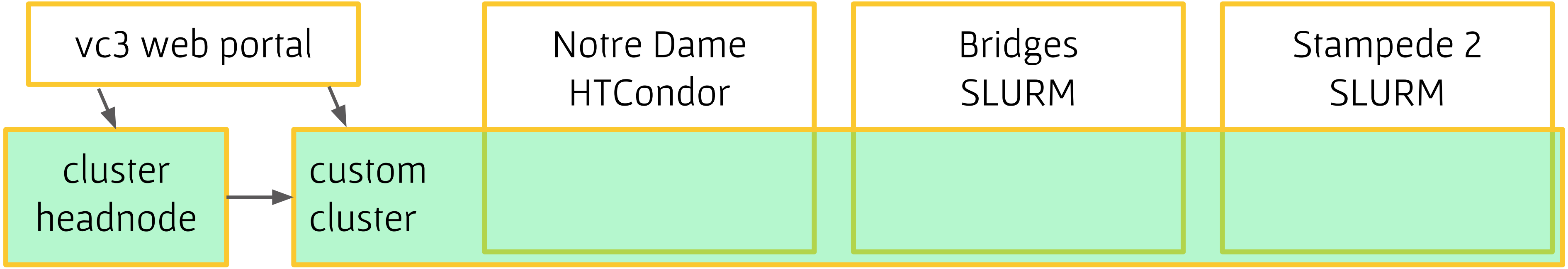
Bridges SLURM

Stampede 2 SLURM

What do we gain?

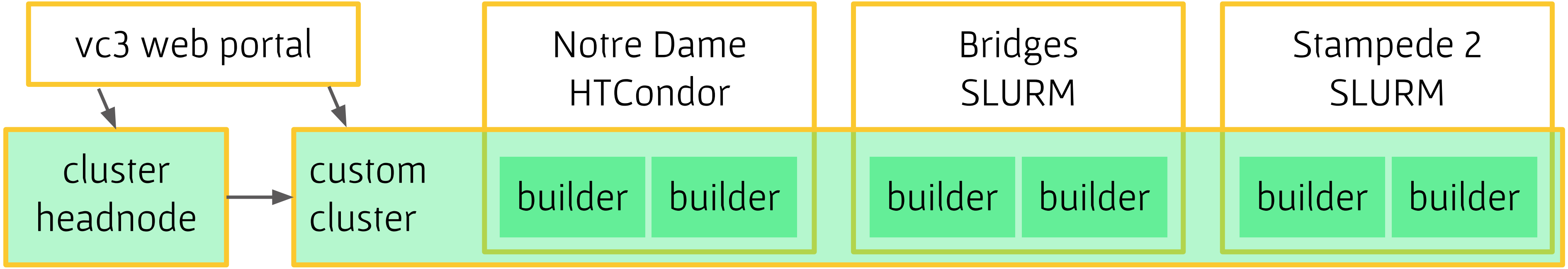
The vc3-builder was written in the context of the **V**irtual **C**lusters for **C**ommunity **C**omputation (VC3) project.

Users go to a **website** and create **short lived clusters** across **heterogeneous resources.**



What do we gain?

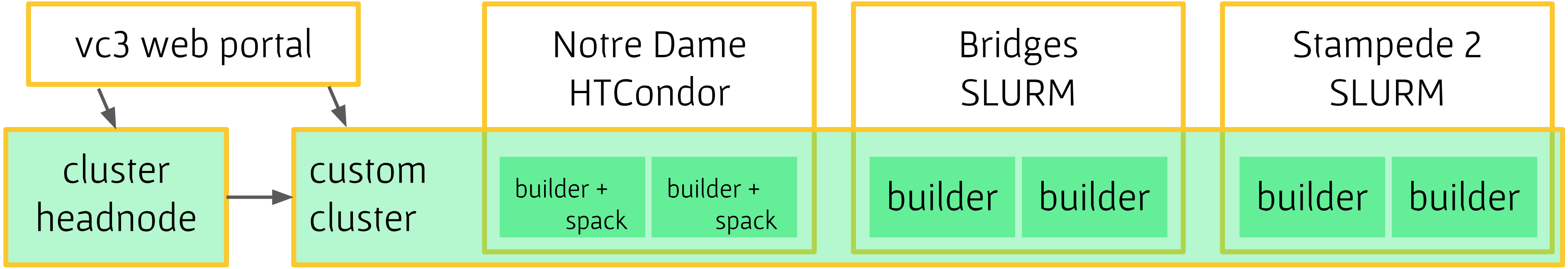
The vc3-builder is deployed as part of job submission to provide consistent software environments.



What do we gain?

The vc3-builder is deployed as part of job submission to provide consistent software environments.

The software environment may be constructed using other tools (e.g. spack)



Another gain: batch system mode

```
./vc3-builder --parallel-mode slurm ...
```

Several vc3-builders running in parallel for the same installation, as dependencies allow.

vc3-builders are submitted as regular batch jobs.

SLURM, HTCondor, WorkQueue, SGE, PBS, and torque are supported.

Only if a shared filesystem is available.

Case study: MAKER

MAKER is a genome annotation pipeline used in bioinformatics.

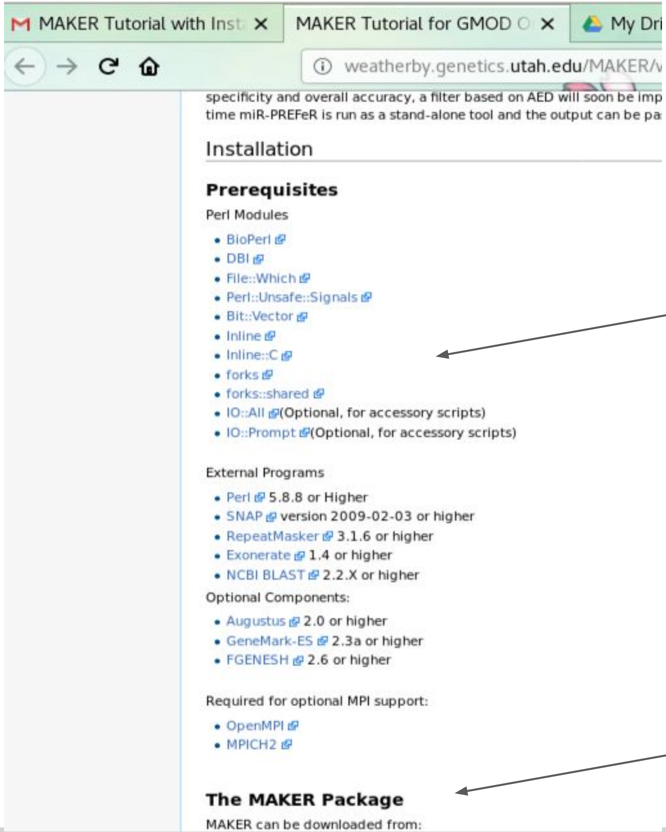
In total, an installation of MAKER consists of **39** dependencies.

The size of the sources is **785M**.
The installation size is **4.2G.**

(see paper for Octave and cvmfs case studies)

MAKER without the vc3-builder

Someone sends you an email with a link with instructions how to install MAKER.


MAKER Tutorial with Inst
MAKER Tutorial for GMOD O
My Dri
weatherby.genetics.utah.edu/MAKER/v

specificity and overall accuracy, a filter based on AED will soon be imp
time miR-PREFeR is run as a stand-alone tool and the output can be pa

Installation

Prerequisites

Perl Modules

- BioPerl
- DBI
- File::Which
- Perl::Unsafe::Signals
- Bit::Vector
- Inline
- Inline::C
- forks
- forks::shared
- IO::All (Optional, for accessory scripts)
- IO::Prompt (Optional, for accessory scripts)

External Programs

- Perl 5.8.8 or Higher
- SNAP version 2009-02-03 or higher
- RepeatMasker 3.1.6 or higher
- Exonerate 1.4 or higher
- NCBI BLAST 2.2.X or higher

Optional Components:

- Augustus 2.0 or higher
- GeneMark-ES 2.3a or higher
- FGENESH 2.6 or higher

Required for optional MPI support:

- OpenMPI
- MPICH2

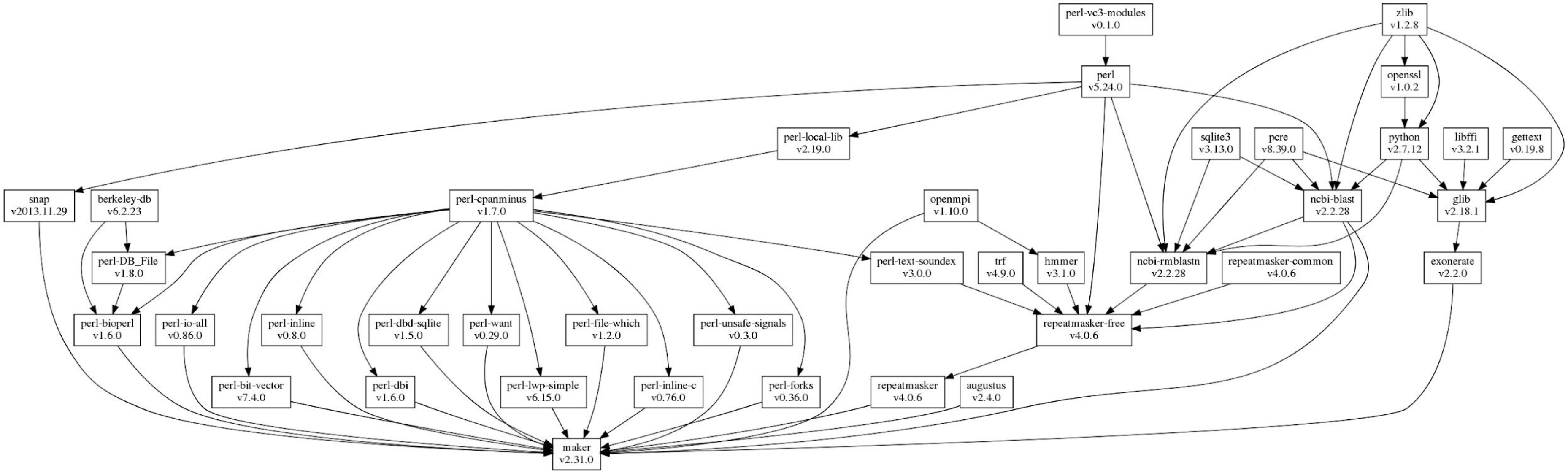
The MAKER Package

MAKER can be downloaded from:

Dependencies of MAKER (themselves have other dependencies)

MAKER proper is over here...

MAKER dependency graph



perl-vc3-modules v0.1.0

zlib v1.2.8

perl v5.24.0

openssl v1.0.2

perl-local-lib v2.19.0

sqlite3 v3.13.0

pcre v8.39.0

python v2.7.12

libffi v3.2.1

gettext v0.19.8

snap v2013.11.29

berkeley-db v6.2.23

perl-cpanminus v1.7.0

openmpi v1.10.0

ncbi-blast v2.2.28

glib v2.18.1

perl-DB_File v1.8.0

perl-text-soundex v3.0.0

trf v4.9.0

hmmer v3.1.0

ncbi-rmblastn v2.2.28

repeatmasker-common v4.0.6

exonerate v2.2.0

perl-bioperl v1.6.0

perl-io-all v0.86.0

perl-inline v0.8.0

perl-dbd-sqlite v1.5.0

perl-want v0.29.0

perl-file-which v1.2.0

perl-unsafe-signals v0.3.0

repeatmasker-free v4.0.6

perl-bit-vector v7.4.0

perl-dbi v1.6.0

perl-lwp-simple v6.15.0

perl-inline-c v0.76.0

perl-forks v0.36.0

repeatmasker v4.0.6

augustus v2.4.0

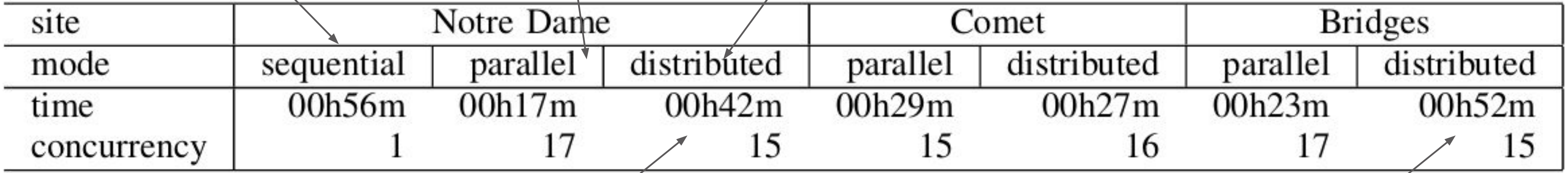
maker v2.31.0

MAKER timings

at headnode, one package at a time

at headnode, several packages at a time

using the batch system, several packages at a time



site	Notre Dame			Comet		Bridges	
mode	sequential	parallel	distributed	parallel	distributed	parallel	distributed
time	00h56m	00h17m	00h42m	00h29m	00h27m	00h23m	00h52m
concurrency	1	17	15	15	16	17	15

mostly waiting in HTCondor queue

mostly waiting in SLURM queue

Conclusions

In a laptop users can easily install software dependencies (e.g. rpms).

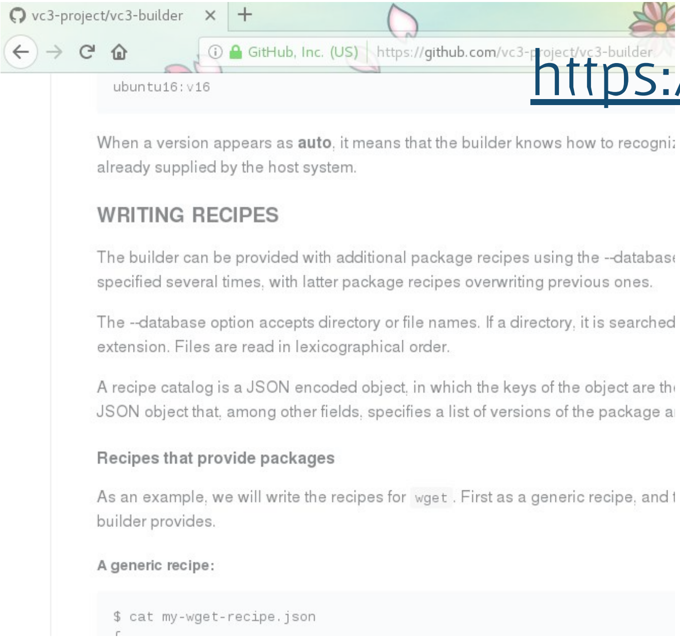
Making those dependencies explicit is hard, but needed in batch/cloud contexts.

Containers by themselves don't help us in making dependencies explicit.

Configure in a laptop, run anywhere in a batch/cloud without sudo, as part of job execution.

Thanks!

btovar@nd.edu



```
ubuntu16:v16
```

When a version appears as **auto**, it means that the builder knows how to recogni: already supplied by the host system.

WRITING RECIPES

The builder can be provided with additional package recipes using the --databas specified several times, with latter package recipes overwriting previous ones.

The --database option accepts directory or file names. If a directory, it is searched extension. Files are read in lexicographical order.

A recipe catalog is a JSON encoded object, in which the keys of the object are th JSON object that, among other fields, specifies a list of versions of the package a

Recipes that provide packages

As an example, we will write the recipes for `wget`. First as a generic recipe, and t builder provides.

A generic recipe:

```
$ cat my-wget-recipe.json
```

https://github.com/vc3-project/vc3-builder

https://www.virtualclusters.org

funding: Department of Energy
NGNS program via grant DE-SC0015711

thanks to the VC3 team,
specially to Kenyi Hurtado
for testing the vc3-builder