Indexing BackPAN

brian d foy
brian@stonehenge.com

April 22, 2008

Stonehenge

- BackPAN is the historical archive of Comprehensive Perl Archive Network (CPAN)
- http://backpan.cpan.org
- 200k number of files, 10 Gb

- **CPAN only has the distributions the authors leave on there**
- **55k distributions, 4 Gb**
- **CPAN tools use an index**

- Perl doesn't have a package manager
- Install a file by putting it in @INC
- No file to distro reverse mapping
- Avoids overwriting by PAUSE indexing and permissions checking
- No version management, multiple versions, author management

- Recreate module installation history
- Start with the files in @INC
- Work backward to distro
- End with a list of distros to install
- Create a MyCPAN with those distros

How PAUSE indexes

- PAUSE accepts uploads from anyone
- Need PAUSE ID, but that's easy to get
- PAUSE does not want to run any code
- Wants author, namespace, version
- It happens in *mldistwatch*

Extract $VERSION

```
next unless /([\$*])(([\w\:\']*)\bVERSION)\b.*\=/;

my $current_parsed_line = $_;
my $eval = qq{
    package ExtUtils::MakeMaker::_version;

    local $1$2;
    \$$2=undef; do {
        $_
        }; \$$2
    };

 $result = eval( $eval );
```

mldistwatch

Stonehenge

- An author has namespace privileges
- Can't upload without permission
- Further uploads have higher versions
- Index failures don't prevent uploads

Extract package

```
if( $pline =~ m{
    (.*)
    \bpackage\s+
    ([\w\:\']+)
    \s*
    ( $ | [\}\;] )
    }x) {
      $pkg = $2;
      }
```

Create an index

- *02.package.details.txt.gz*

```
DBI   1.604  T/TI/TIMB/DBI-1.604.tar.gz
```

- Has only latest distro
- This isn’t a magic file
- But CPAN tools use it

How I Index

- Don’t care about permissions
- PAUSE has already filtered
- Run in virtual machines
  - no network connections
  - mount BackPAN readonly
  - if it blows up, so what

- **Don't trust anything**
  - **Not *META.yml*, *Makefile.PL*, *Build.PL***
- **Run the build file, look in *blib***
- **Extract *blib* file list, file meta data, namespaces, and versions**
- **Extract anything else I can**
  - **Dependencies**

- Mostly automated
- Use one set-up, index
- See what fails
- Try another to get more
- Try different methods

- Right now, I just want the data
- Distribute data in many forms
- People can use it how they like
- Keep up with CPAN

Mechanics

Unpack dist

- Archive::Extract
- Automatically dispatches

```
my $extractor = eval {
   Archive::Extract->new( archive => $dist ) };
my $rc = $extractor->extract( to => $unpack_dir );
my $type = $extractor->type; $ tgz, etc.
```

Fork

- Each dist gets it’s own process
- Compartmentalize
- Parallelize
- alarm-ize

Extract versions

- Module::Extract::VERSION
- Same as PAUSE, but in a module

```
Module::Extract::VERSION
    ->parse_version_safely( FILE );
```

- Can be changed later

Guess build system

- Distribution::Guess::BuildSystem
- Try different techniques
- Disable `auto_install`
- Create *blib*

```
use Distribution::Guess::BuildSystem;
my $guesser =
   Distribution::Guess::BuildSystem->new(
     $dist_dir );

if( $guesser->uses_makemaker ) { ... }
elsif( $guesser->uses_module_build ) { ... }
elsif( ... ) { ... }
```

Record meta data

- Filenames
- File size
- MD5 digests
- Source control keywords
- PPI cache?

Extract packages

- All packages

```
use Module::Extract::Namespaces;

# in list context, extract all namespaces

my @namespaces =
Module::Extract::Namespaces

    ->from_file( $filename );
```

- Assume first package is main one

Use PPI

```
my $package_statements = $Document->find(
  sub {
  $_[1]->isa('PPI::Statement::Package')}
  );
my @namespaces = map {
   /package \s+ (\w+(::\w+)*) \s* ; /x;
   $1 } @$package_statements;


     }
```

Record as YAML

- Changing too much for a database
- Easier to look at
- Can import later
- Can hand edit to correct

Notice errors

- Some distros don’t build
  - Perl versions
  - Perl compilation options (threads)
  - OS dependencies
  - missing libraries

Modify system

- Find out what doesn’t work
- Fix the indexer for it
- Make a special case

Conclusion

- Index all of BackPAN
- Modularize bits of PAUSE
- Redistribute the data
- Create custom CPAN versions