

# PL/Parrot and PL/Perl6

## Parrots and Butterflies in your Database

Jonathan "Duke" Leto

# Parrot Virtual Machine

- Process (Application) Virtual Machine
- Register-based
- Continuation Passing Style
- Design Goals
  - Pluggable
  - Interoperable
  - Dynamic
- 2.7.0 "Australian King" just released
- Undergoing lots of optimizations

## Rakudo Perl 6

- Most active implementation of Perl 6
- Implements  $\sim 80\%$  of the spec
- Currently uses Parrot as a backend, but plans to support others
- Stable release series is Rakudo Star, dev release every month
- Getting faster due to optimizations in Parrot

# Why Embed Parrot VM in PostgreSQL?

- Procedural/PostgreSQL Languages (PL's) are hard to write and maintain
- PL/Parrot does the hard work, HLLs benefit with much less effort
- Platform independent, fast<sup>†</sup>, stored procedures

PL/Parrot and  
PL/Perl6

Parrots  
and

Butterflies  
in your  
Database

@dukeleto

# History of PL/Parrot

## Current Features

- PL/PIR and PL/Perl6
- Pass and return basic datatypes
- Basic security model (Don't do that)
- Growing Test Suite
- Enthusiastic and friendly community
- Coolest Feature: Use Perl 6 grammars in PostgreSQL!

# Things that creak

- Documentation - <http://pl.parrot.org>
- SPI - branch being worked by cxreg++, elog works!
- Triggers
- SETOF - branch with some tests
- Parrot Bugs
  - IMCC Syntax Errors
  - Security API
  - Extend/Embed API

# PL/PIR Example 1

```
CREATE FUNCTION test_float_add(float) RETURNS float AS $$  
    .param num x  
    x += 5  
    .return(x)  
$$ LANGUAGE plparrot;
```



# Installing/Testing PL/Parrot

- Install Rakudo or Parrot
- `# parrot_config` needs to be in your `$PATH`
- One of:
  - `git clone git://github.com/leto/plparrot.git`
  - `wget http://icanhaz.com/plparrot0.20`
  - `cd plparrot`
  - `export PGPORT=5555 # if necessary`
  - `make install installcheck # might need sudo`
  - `make test test_plperl6 # PL/PIR + PL/Perl6 tests`

## Future Goals

- Easy onramps to add new languages to PL/Parrot
- Framework for DSL's
- Allow various PL's to communicate
- Freeze/thaw subtransaction-level states

## Get involved!

- Try PL/Parrot on your system and submit detailed bug reports
- Fork on github and hack on stuff!
- Help with GitHub Issues
- #plparrot on freenode
- <http://pl.parrot.org>
- <http://groups.google.com/group/plparrot>

# Thanks

- PL/Parrot team:
  - David Fetter, David E. Wheeler, Joshua Tolley, Daniel Arbelo Arrocha, Dave Olszewski
  - AKA davidfetter++, theory++, eggyknap++, darbelo++, cxreg++
- Everyone working on Parrot Virtual Machine, Perl 6 and PostgreSQL
- Especially (Moritz Lenz) moritz++, (Peter Lobsinger) plobsing++ and (Stephen O'Rear) sorear++, for great advice and help

# Resources

- <http://pl.parrot.org>
- <http://github.com/leto/plparrot>
- <http://parrot.org>
- @parrotvm / !parrot on twitter/identi.ca
- @dukeleto / @leto on twitter/identi.ca