Parrot and Perl 6

Simon Cozens

NetThink Open Source Consultants

The Great Decision

- TPC 4, 2000, Monterey.
- Emergency meeting called by Chip Salzenburg
- Interrupted by Jon Orwant
- Complete redesign proposed, and accepted at P5P meeting
- Roles distributed

The rewrite

- A complete, ground-up rewrite
- Redesign of language and internals
- Not afraid to break backwards compatibility
- This did not please everyone
- Least of all me...

The process

- RFCs: Community suggestions for redesign
- PDDs: "Official" implementation documents
- Apocalypses: Revelations of Larry’s design
- Exegeses: Damian’s user-oriented explanations
- Metamorphoses: (to come) "Translations" of the Perl Cookbook

The Parrot Sketch

- Kick-ass AFJ, needing a cast of thousands
- Secret, serious idea...
- ... to produce a dynamic, "shared" bytecode interpreter
- to rival the .NET CLR
- Plan solidified after TPC 2001 at ESR’s house

Relationship to .NET

- .NET CLR only likes things that look like C#
- Perl 6 could not look much less like C#
- Static versus dynamic
- Impedence mismatch
- Same goes for mono

The implementation

- Dan Sugalski producing an internals design
- Your humble servant, chief implementor
- Six CVS committers
- 30k lines of code so far
- More needed

Overall design

- Bytecode interpreter
- Fed bytecode from compiler front-end
- Tight loop executing the bytecode
- It’s register based
- 32 x 4 registers
- Handles GC, finalization, ...

Language Independence

- If we’re to be truly language-independent...
- can’t make assumptions about variable behaviour
- But still want macro-ops
- So what to do?

PMCs

- Parrot (originally "Perl") Magic Cookies
- "Magic" because we don’t want you to poke inside them
- Doesn’t that sound like an object to you?
- Language-specific behavior implemented in methods and classes

PMCs and vtables

- Methods are vtables, structs of function pointers
- Dan came up with the initial design
- Then I noticed that Python does exactly this
- So I stole it

Other cool things

- user-defined opcodes
- sifo-like (non)standard IO subsystem
- Proper threading support!

Current status

- Initial release 10/09/2001
- Read/write bytecode, simple operations on registers
- 0.0.2 release 05/10/2001
- Portability release
- Core platforms: Linux, FreeBSD, Cygwin, Tru64, Win32, OpenVMS, Solaris Sparc
- Minilanguages

Future status

- 0.0.3 release was scheduled for last week, but I didn't get enough coding time
- Will support Perl scalar PMCs
- 0.0.4 release will support simple arrays and hashes
- 0.0.5 release will support subroutines, lexical variables and more