Erlang Training and Consulting Ltd

Erlang Testing and Tools Survey

Aniko Nagyné Víg and Tamás Nagy

Victoria, British Columbia, Canada, 27 Sept 2008

Agenda

- Introduction
- Research method for the Survey
- Erlang tools
- Tool requirements for commercial products
- Conclusion

Introduction

What we are doing and why

Goals and tasks

- Market analysis of Erlang and non-Erlang Tools
  - as a workpackage of the ProTest project
- Compare the Tools available in Erlang to ones in other languages (not only functional languages)
  - Details in the paper
- Strength and Weakness analysis of the Tools

Erlang

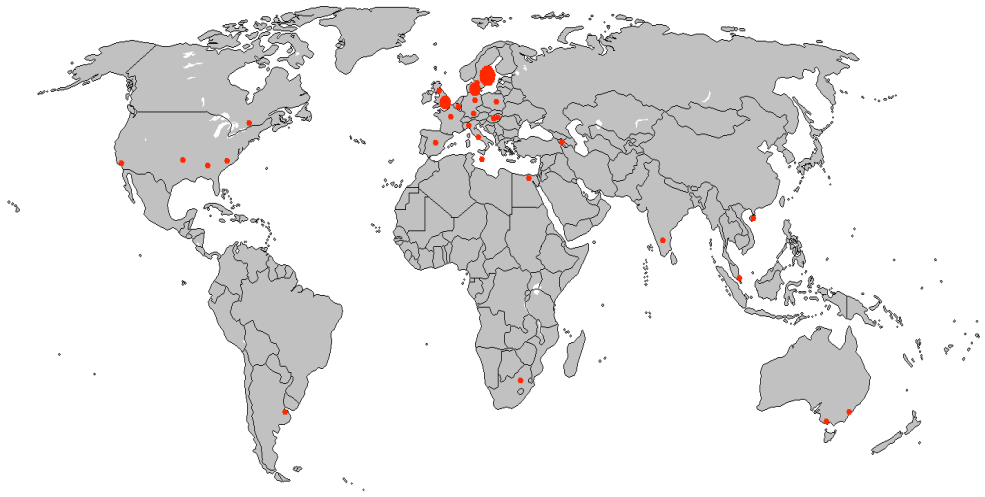
Research method for the Survey

Online survey done to gather data about the usage and spread of Erlang tools and applications in the community

Research method

- Published an online survey
- Advertised it by email:
- Erlang Questions, approx. 1000 users
- Erlang Training and Consulting Newsletter list, approx. 1000 users
- Smaller Erlang related mailing lists
  - Trapexit User Group, 500 users
  - ProTest Mailing List, 50 users
  - London / Stockholm Erlang User Groups, 100 users
- 200 direct emails to relevant contacts at ETC
- Merged with the main survey after the results were similar
- 40-45% of total(200) responses were from developers

Geographical diversion



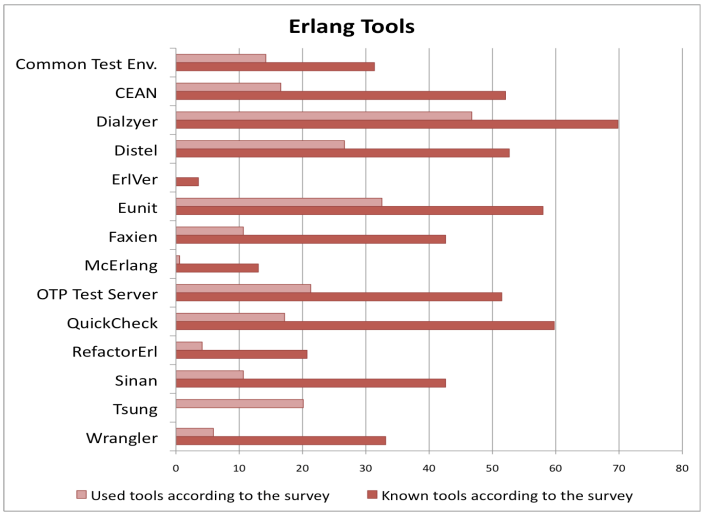
Structure of the survey

The survey contained 20 questions about:

- The development environment of Erlang users
- Usage and knowledge of existing Erlang tools & applications
- Submitter's job role and Erlang background
- Identify common processes to improve tool support

Erlang tools

Popularity of Erlang tools among the community



Erlang tools

Testing tools

EUnit

- Open source light weight test server
  - Unit testing framework
- Easy to write test cases
- Fast test cycle
- Test results presented accurately
- Allows
  - implementation of test generating functions
  - Execution of the test representation

OTP Test Server and Common Test Environment

- Test suite execution environment based on Erlang
  - Supports regular automated testing
- Portable test server for application testing
- Common Test:
  - a framework based on the OTP Test Server application

OTP Test Server and Common Test Environment 2

- Suitable for
  - black-box (on any target through O&M interfaces)
  - white-box testing
- Provides code coverage analysis

According to the survey, “the set up of the test environment is complex, suffers lack of documentation and its strict regulation about name conventions makes it hard to use and adopt”

QuickCheck

- Commercial tool for property and model based testing
- Tests against running applications
- Different levels of testing
  - From unit testing to system testing
- Built in automated test case simplification called '**shrinking**'
- Efficient for testing complex systems in early development phases
  - Allows to test against formal specifications

- Limited user feedback

- Hard to learn the logic for writing test cases, even for Erlang developers

Erlang tools

Tools for development activities other than testing

Refactoring tools

Refactoring means improving the quality of code through re-writing without changing its external behaviour.

RefactorErl

- Has an installation tool for Windows and source package for Linux & MacOS
- Supports seven transformations

Wrangler

- Integrated into both Emacs and Eclipse
- Supports seven transformations and two search options

Analysis tool

Dialyzer is a static analysis tool for detecting discrepancies in the source or byte code files automatically

- Typical errors detected by the tool
  - Type errors
  - Redundant tests
  - Unsafe virtual machine byte code
  - Unreachable code

Model checker

Etomcrl is a model checker for Erlang. It checks state equality, but the input language is rather restricted

McErlang is a model checker for verifying distributed Erlang programs

- McErlang has support for
  - General process communication
  - Node semantics
  - OTP component libraries
  - Fault detection and tolerance through process linking.

Automated build tools and package management

Faxien is a package management tool which helps to find, install and publish OTP applications.

- **Sinan** is a build system for OTP projects.
  - Compiles and builds OTP applications
  - Builds documentation
  - Runs dialyzer
  - Checks unit tests
  - Produces reports
  - Handles application dependencies.

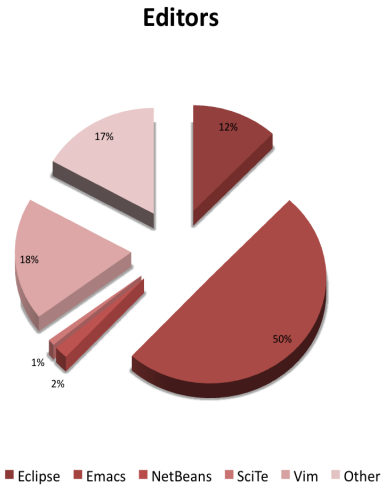
Erlang tools

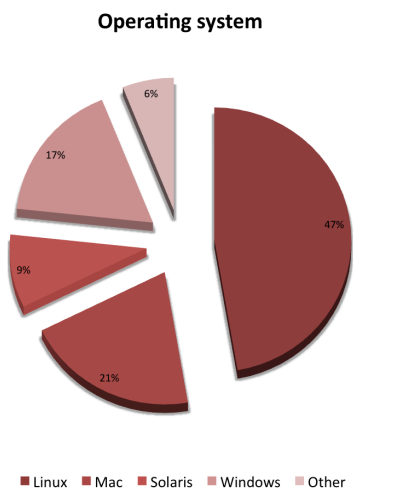
Usability of the tools

User friendliness

- The interface should be easy to use
- Well integrated to the development environment
- The layer of integration can be on the following levels:
  - Editors with plugins
  - OS command line
  - Developing software interfaces (Erlang shell/CLI)

Editors and OS used in the community





Erlang tools

Open problems identified by the survey

Open problems identified by the survey

- Weaknesses of most Erlang tools and projects were found to be
  - Lack of documentation
  - Lack of examples and tutorials
  - Incomplete and untested tools
- Design issues included
  - Badly layered software
  - Not extensible and not structured
- Doubts about sustainability
- Hard to install and use
  - Especially for non Erlang users
  - Extensive manual configuration required

Missing functionality

- No tools for stub generation
- Testing tools lack high quality results display
  - Web interface or dashboard with statistics and graphs
- Load testers are not available for all requirements
  - Especially state based protocols
- Continous integration
- Hooks towards version control systems
  - Integrated into a general framework
- A complete framework that integrates different tools
  - Compile, unit test, system test, refactor, xref, dialyzer & tsung

Tool Requirements for Commercial Products

Tool requirements for commercial products

- Ease of use
- Active Support
- Good documentation
- Examples
- Ease of comprehending the results

Conclusion

Conclusion

What are the key factors for building a successful Erlang tool?

- Reliable software
- User friendliness
- Good documentation
- Support
- Well promoted

Thank you for your attention!

Any questions?