dBabble*: Cross-platform Language Interfacing To Facilitate Open Science Collaborations

Gil Alterovitz[1,2,3,4], Kshitij Marwah[5], Paresh Malalur[2], Anshuman Aggarwal[5], and Marco F. Ramoni[1,3,4]

[1] Division of Health Sciences and Technology, Harvard University/Massachusetts Institute of Technology, Cambridge, MA, USA. [2] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA , USA. [3] Children's Hospital Informatics Program, Boston, MA, USA. [4] Harvard Partners Center for Genetics and Genomics, Harvard Medical School, Boston, MA , USA. [5]Department of Computer Science, Indian Institute of Technology Delhi, Delhi, India.

Abstract

- Open science promises increased productivity via communication and collaboration in what is now a global research community.

- Bioinformatics and biocomputing open science research is unique in that it joins very different disciplines such as biology, mathematics, engineering, and computer science together.

- Researchers in bioinformatics thus typically come with their own computer languages. Perhaps more importantly, each researcher knows and can access vast libraries of code within his/her specialty's community (written in his/her native language).

- Current programming models for cross-language use (e.g. swig, http://www.swig.org/) assume one high level language for prototyping and a low level one for optimizing code.

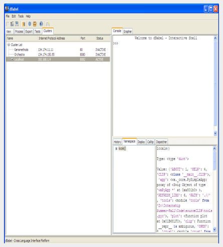
- dBabel is open source and available via GPL at: http://bcl.med.harvard.edu/proj/dbabel

Introduction

- dBabel provides a novel framework for communication between various languages and platforms.

- dBabel automatically detects and integrate code from different languages spread over different platforms and locations- so that they can communicate easily with one another.

- dBabel offers the user an intuitive graphical interface with a powerful interactive shell to manage various clusters spread in various facilities all around the world.



dBabel
File Edit Tools Help
View Process Export Tests Clusters
Console Grapher
History Namespace Display Calltip Dispatcher

Name	Internet Protocol Address	Port	Status
Cluster List			
Genomethods	134.174.11.11	80	INACTIVE
Orchestra	134.174.150.55	8080	INACTIVE
Localhost	192.168.1.4	8082	ACTIVE

```
Welcome to dBabel - Interactive Shell
>>>
```

```
locals()

Type: <type 'dict'>

Value: {'ABOUT': 1, 'HELP': 4,
'CLIP': <class '__main__.CLIP'>,
 'app': <wx._core.PySimpleApp;
proxy of <Swig Object of type
'wxPyApp *' at 0xa501b0> >,
'REFRESH_LIBS': 6, 'PATH': './/'
, 'tools': <module 'tools' from
'D:\Internship
Summer-Fall\Code\sourceCLIP\tools
.pyc'>, 'plot': <function plot
at 0x01B681F0>, 'clip': Function
 __repr__ is ambiguous, 'OPEN':
```

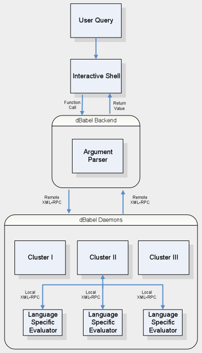
dBabel - Cross Language Interface Platform

Methods

- dBabel uses XML-RPC, a simple yet robust remote procedure calling mechanism. It uses HTTP as the transport and XML as the encoding, thus allowing complex data structures to be transmitted, processed and returned.

- The user query is parsed by the dBabel backend and a list of clusters running the dBabel daemon service are contacted to check for the existence of the function.

- On finding the required function, the language specific XML-RPC service is spawned in the cluster, the function is evaluated, and the result sent back to dBabel.



User Query

Interactive Shell

Function Call

Return Value

dBabel Backend

Argument Parser

Remote XML-RPC

Remote XML-RPC

dBabel Daemons

Cluster I

Cluster II

Cluster III

Local XML-RPC

Local XML-RPC

Local XML-RPC

Language Specific Evaluator

Language Specific Evaluator

Language Specific Evaluator

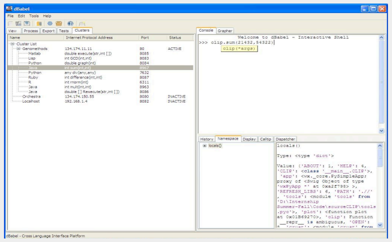
* Formerly called: dBabble

Results

- dBabel is implemented in python and currently supports Python, Java, Lisp, Matlab, R, Perl, C, C++, and Ruby.

- dBabel interfaces with the outside world via a powerful and interactive GUI shell. Among its many features are smart syntax highlighting and auto-detection of code from different languages.

- dBabel also supports Application Programming Interfaces (API's) in Python, Java, Matlab, R, and Lisp- enabling virtually any language to communicate with one another.



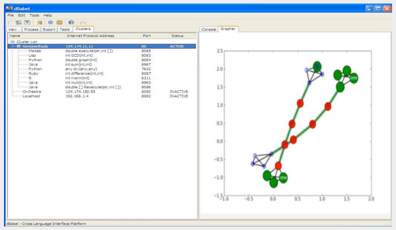
dBabel
File Edit Tools Help
View Process Export Tests Clusters
Console Grapher
clip(*args)
History Namespace Display Calltip Dispatcher
locals()

Name	Internet Protocol Address	Port	Status
Cluster List			
Genomethods	134.174.11.11	80	ACTIVE
Matlab	double execute(str,int [ ])	8085	
Lisp	int GCD(int,int)	8083	
Python	double graph(int)	8084	
Java	int sum(int,int)	8967	
Python	any div(any,any)	7632	
Ruby	int difference(int,int)	8087	
R	int rnorm(int)	6311	
Java	int mult(int,int)	8963	
Java	double [ ] Rexecute(str,int [ ])	8086	
Orchestra	134.174.150.55	8080	INACTIVE
Localhost	192.168.1.4	8082	INACTIVE

```
Welcome to dBabel - Interactive Shell
>>> clip.sum(21432,54322)
```

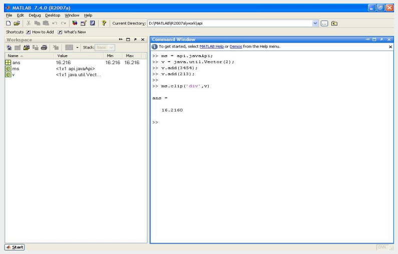
```
locals()

Type: <type 'dict'>

Value: {'ABOUT': 1, 'HELP': 4,
'CLIP': <class '__main__.CLIP'>,
 'app': <wx._core.PySimpleApp;
proxy of <Swig Object of type
'wxPyApp *' at 0xa2f798> >,
'REFRESH_LIBS': 6, 'PATH': './/'
, 'tools': <module 'tools' from
'D:\Internship
Summer-Fall\Code\sourceCLIP\tools
.pyc'>, 'plot': <function plot
at 0x01B69270>, 'clip': Function
 __repr__ is ambiguous, 'OPEN':
```

dBabel - Cross Language Interface Platform



Name	Internet Protocol Address	Port	Status
Cluster List			
Genomethods	134.174.11.11	80	ACTIVE
Matlab	double execute(str,int [ ])	8085	
Lisp	int GCD(int,int)	8083	
Python	double graph(int)	8084	
Java	int sum(int,int)	8967	
Python	any div(any,any)	7632	
Ruby	int difference(int,int)	8087	
R	int rnorm(int)	6311	
Java	int mult(int,int)	8963	
Java	double [ ] Rexecute(str,int [ ])	8086	
Orchestra	134.174.150.55	8080	INACTIVE
Localhost	192.168.1.4	8082	INACTIVE

dBabel - Cross Language Interface Platform



MATLAB 7.4.0 (R2007a)
File Edit Debug Desktop Window Help
Shortcuts How to Add What's New
Workspace
Stack: Base
Command Window
Start
OVR

Current Directory: D:\MATLAB\R2007a\work\api

Name	Value	Min	Max
ans	16.216	16.216	16.216
ms	<1x1 api.javaApi>		
v	<1x1 java.util.Vect...		

To get started, select MATLAB Help or Demos from the Help menu.

```
>> ms = api.javaApi;
>> v = java.util.Vector(2);
>> v.add(3454);
>> v.add(213);
>>
>> ms.clip('div',v)

ans =

   16.2160

>>
```

Conclusions

- dBabel's combines languages and platforms together within a single framework.

- Most of the major languages used in bioinformatics and biocomputing are currently supported allowing researchers all over the world to communicate seamlessly.

- dBabel has already been used to foster both international open science collaborations as well local initiatives.

- Current directions for dBabel include: recruiting social networks to share computational resources, integrating more languages, and creating additional API's.

- For more information on dBabel, please see: http://bcl.med.harvard.edu/proj/dbabel

References

1. Dave Winer. XML-RPC Specification, http://www.xmlrpc.com/spec, June 1999.

2. Simplified Wrapper And Generator (SWIG), http://www.swig.org/

3. Mark Allman. An Evaluation Of XML-RPC, March 2003.

4. Mark Conrad, Tim French. A Gentle Transition from Java Programming to Web Services using XML-RPC.

5. Simon Urbanek. Rserve A Fast Way To Provide R Functionality to Applications. Proceedings of the 3rd International Workshop on Distributed Statistical Computing. March 20-22, 2003.

Acknowledgements

This work was supported under NLM grant 1K99LM009826 and PSB/ Open Science Workshop.

PARTNERS HEALTHCARE

BWH

Biomedical Cybernetics
Laboratory

bcl.med.harvard.edu