A Window to Galaxy

Liram Vardi and Amir Ben-Dor
Agilent Labs, Tel Aviv
Galaxy is a Linux Fan

• Galaxy is a great framework for Bioinformatics researchers.

• Which OS? Windows or Linux?
  – Great community support for Linux based Galaxy
  – Many RNA-seq analysis tools available for Linux
  – NO support for Windows-based Galaxy

We chose Linux 😊
Mixed Workflow

• There are many tools that are developed and compiled for Windows, and therefore cannot be integrated into Galaxy workflows.

• Galaxy needs to embrace all developers (even if they made a mistake with the OS 😊).
Galaxy @ Agilent

Within Agilent we have both Linux and Windows developers.

• Genomics:
  – RNA-seq
  – Most of the tools are executed on Linux platform

• Proteomics:
  – Mass spectrometry data analysis
  – Agilent Spectrum Mill (Windows based)

The Problem

• Executable that refers to “Dynamic Link Libraries” (DLLs) will execute only on windows.

• Code written in .NET languages (C#, VB.net, etc) is compiled to a windows specific runtime
Do we really have to choose?
Existing Solution I - Mono

• Mono is a free and open source project led by Xamarin.
• Able to run Microsoft .NET applications cross-platform.
• Includes .NET Framework-compatible set of tools including:
  – C# compiler
  – Common Language Runtime
Existing Solution II - Wine

• Wine is a open source compatibility layer.
• Aims to allow computer programs written for Windows to run on Unix-like operating systems.
• It duplicates functions of a Windows computer by providing alternative implementations of the DLLs.
Drawbacks (personal experience)

Both are NOT supported by Microsoft Mono:
- Partial support for the .NET features, such as WPF, WCF, etc.
- In many cases a Mono recompilation of the source code is needed.
  - Additional work
  - Not always possible due to unavailability of the source code.

Wine:
- Only an old version exists at the Red Hat distribution.
- Problematic to install on remote machine without a display adaptor.
- Requires Mono to execute .Net programs.
Our Proposed Solution

**Goals:**

- Enable running any Windows application from Linux
- Integrate Linux and Windows tools within the same galaxy workflow.
- Transparency of the integration to the Galaxy end user.
Example Hybrid Galaxy Workflow

This step is done on a Windows machine.
A Window to Galaxy

Method:
Create a collaboration (relationship...) between two separate machines (Linux client and Windows server)
• The “server” can be also a guest VM.
Galaxy Layers Model

User interface (history)

my_data.fastq

results.fastq

Galaxy

Data_012.dat

Data_013.dat

Your Linux tool

$ your_tool data_012.dat data_013.dat
However, with a Windows-based Tool
Window2Galaxy Implementation

- Samba
- VM share directory

HTTP requests
WinTool execution

“WinTool $input $output”

HTTP request: “Execute WinTool $input $output”

WinTool <Win_path>/\input WinTool <Linux_path>/\output

Window2Galaxy

$Input / $output

$Input
WinTool Execution

HTTP request: Is job done?
Respond: Yes/No

$output

Standard Output Stream
Standard Error Stream
The Gory Details – Server Side

• Web service implemented on ASP.NET (Active Server Pages)
• Embedded in IIS7
• Window2Galaxy can execute any command (security breach?)
  – Job executed under IIS Worker (limited permissions) account
  – Users cannot integrate new tools and XML configuration without Galaxy administrator permissions
Easy Installation

- Prerequisites:
  - Server side: IIS7 installed
  - Setting up a shared directory
- Client installer (Linux side).
- Server Installer:
Summary

• Galaxy is a great framework (already said?).
• There are business and research needs to incorporate Windows-based tools into Galaxy workflows.
• Benefits of Window2Galaxy:
  – Enables running any Windows tool in a native fashion.
  – Totally transparent to the Galaxy end user
  – General mechanism for running Windows tools from Linux (not just for Galaxy users)
Agilent Labs - Israel
Thank you!