2011

The Cloud Agnostic e-Science Analysis Platform

Ajith Harshana Ranabahu
Wright State University - Main Campus

Paul E. Anderson
Wright State University - Main Campus

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

Repository Citation
Ranabahu, A. H., Anderson, P. E., & Sheth, A. P. (2011). The Cloud Agnostic e-Science Analysis Platform. *IEEE Internet Computing, 15* (6), 85-89.
https://corescholar.libraries.wright.edu/knoesis/609

Editor: Schahram Dustdar • dustdar@infosys.tuwien.ac.at

The Cloud Agnostic e-Science Analysis Platform

Ajith Ranabahu • *Kno.e.sis, Wright State University*

Paul Anderson • *College of Charleston*

Amit Sheth • *Kno.e.sis, Wright State University*

The amount of data being generated for e-Science domains has grown exponentially in the past decade, yet the adoption of new computational techniques in these fields hasn't seen similar improvements. The presented platform can exploit the power of cloud computing while providing abstractions for scientists to create highly scalable data processing workflows.

The recent flood of Web-enabled and Web-based tools for e-Science has provided scientists with a wide array of new methods for collecting, reporting, analyzing, and sharing their data. These Web-based technologies have demonstrated the potential to enable broader collaboration and facilitate the sharing and reuse of experimental data. The wide availability of quality datasets and tested process flows is indeed a welcome addition for e-Scientists, but challenges remain. In scientific domains such as bioinformatics, we can view the lack of standardization for scientific methods, algorithms, and data sharing as the greatest obstacle to broader adoption of new computational methods.

Many e-Science applications, including workflows, have high-performance computation requirements. Cloud computing is increasingly a natural choice for such high-demand computing tasks because minimal upfront investments in computational resources are required. Using computing clouds also avoids the complications of provisioning for periodic peak usage, frequent software and hardware updates, and the need for trained IT staff. However, these advantages are offset by new challenges. The need to program for a specific cloud platform, also called vendor lock-in, is a major hindrance to the widespread adoption of clouds, especially in scientific domains.

We present SCALE, an analysis platform for e-Science experimental data. SCALE supports

- easy collaboration across a community of scientists with minimal operational support from IT and computer scientists; and
- the use of cloud computing resources (private or public) in a platform-agnostic manner to provide high-performance computing when applicable.

Metabolomics — which studies the chemical processes of metabolites — is the first e-Science domain in which we've applied SCALE. Metabolomics is a relatively young research area that requires intensive signal processing and multivariate data analysis to interpret experimental results that require quantifying hundreds of metabolite levels for each sample analyzed. Regardless of the data collection method, metabolomics experiments require substantial computational and statistical support, and researchers must select from among myriad visualization and multivariate statistical techniques for exploration and analysis. We're currently using SCALE tools to support collaboration between Kno.e.sis center computer scientists and biological scientists at Wright State University and the US Air Force Research Laboratory.

1089-7801/11/$26.00 © 2011 IEEE

Here, we discuss the principles behind the SCALE approach and address the challenges faced in creating workflows that scientists can deploy on their cloud of choice. We've been able to provide sufficient abstractions for domain scientists, letting them create their own workflows for the cloud.

Traditional Workflows in the Metabolomics Domain

Although many successful generic workflow languages and tools exist (such as Taverna [www.taverna.org.uk] and Kepler [https://kepler-project.org]), we've observed that the nature of the typical processing task in metabolomics and similar e-Science domains makes it difficult to use an off-the-shelf workflow solution. Additionally, scientists face new challenges if they wish to use highly attractive cloud infrastructure to deploy their workflows. Three aspects are important in this context.

First, scientists often perform analysis over datasets of various sizes. These can range from a few hundred Kbytes to a few hundred Mbytes. Scientists need a flexible approach that lets them use computing resources appropriate to the data size.

Second, many institutions that perform metabolomics analysis must process and keep data in-house for legal reasons. This limits the processing options available to scientists and makes service-based workflow solutions difficult or impossible to use.

Finally, some bioinformatics institutions have their own type of high-performance computing infrastructure or subscription on a cloud infrastructure. The differences between these computing resources ultimately result in highly platform-specific implementations. The disparity of these implementations becomes an issue during inter-lab collaborations.

Given these challenges, we designed SCALE from the ground up to have the following features:

- Provide domain-specific abstractions, comprehensive to domain scientists; such abstractions provide easier collaborations and also let scientists conveniently create domain-specific workflows. Off-the-shelf workflow solutions provide only workflow-specific abstractions and don't provide an advantage in a domain-specific context.
- Be platform-agnostic – that is, scientists can convert the workflows developed using SCALE to run on any applicable platform. This lets scientists use the appropriate computation resource, depending on their processing requirements and the resource's availability. Traditional workflow solutions have specific implementations (*workflow engines*) that run the workflow. There's no concept of supporting multiple platforms; instead, resources external to the platform are accessed as services. SCALE departs from this approach by getting the code to the data, rather than bringing data to the code.
- Have minimal local tooling requirements; many traditional workflow solutions require installing bulky composer tools and numerous plug-ins, an extra burden for a scientist whose primary requirement is to efficiently analyze a dataset.

In the next section, we examine the abstractions used in SCALE.

Abstractions for Metabolomics Workflows

Our work's driving principle is providing abstractions over the typical tasks encountered in a metabolomics workflow. These abstractions are the primary building blocks in creating cloud-agnostic implementations.

We created a set of operator families – that is, groups of operations that are related – to support these abstractions.[1] For example, *normalization* is an operator family and *sum normalization* is an operator under the normalization family. The presence of such operators lets scientists express statistical workflows as platform-agnostic mathematical equations or pseudo code representations that are comprehensive to a wide range of metabolomics researchers.

Consider the statistical workflow of sum normalizing, quantification by binning, and then auto scaling a dataset. This workflow contains examples of routine but nontrivial statistical operations encountered in metabolomics data analysis. We can express this via a mathematical equation:

$$Output = Sauto\ (Qub(Nsum(Input)))$$

as well as a pseudo code representation:

```
data=readFile(File)
normalized= N_sum (data)
binned= Q_ub (normalized)
scaled=S_auto (binned)
writeFile(scaled)
```

An important development in SCALE is the formation of a *domain-specific language* (DSL). A DSL is a programming language or executable specification language that offers, through appropriate notations and abstractions, expressive power focused on and usually restricted to a particular problem domain.[2] Unlike general-purpose programming languages, DSLs are applicable only in their supported domains. Despite this constraint, a large number are in use today, applied to various domains of different granularity.

For SCALE, we created a DSL that closely follows the operators. The following script, written using the SCALE DSL, illustrates the example discussed earlier of loading a dataset and performing three statistical operations over it:

```
# load data
original_data = load_data
  (:raw_data_file,
  {:format => "csv"})
# sum normalize
normalized = sum_normalize
  (original_data)
# binning
binned = uniform_binning
  (normalized)
# Auto scale
scaled = auto_scale(binned)
# write the file
store_data(:scaled, scaled)
```

This script is almost the same as the pseudo code representation, albeit more formal and well-formed. The operator names represent familiar concepts to a domain scientist, giving him or her an immediate understanding of the process flow.

We developed our abstractions based on a well-published metabolomics study.[3] In this study, researchers administered single doses (0.1 to 100 mg/kg of body weight) of an α-naphthylisothiocyanate (ANIT) — a common model of hepatic cholestasis — orally in corn oil to male Fischer 344 rats. The researchers analyzed urine samples via 1H-NMR spectroscopy and processed (sum normalized) and adaptively binned the spectra to reduce dimensionality. After auto scaling, they used an *orthogonal projections to latent structures* (OPLS) methodology to analyze multigroup complex datasets to study complex experimental designs. This methodology allowed scientists to encode the multiple dose and time combinations and directly compare them for novel biomarker identification.

The current set of implemented operators cover the workflow used in this study. Only some operator implementations are cloud-capable. Those that aren't can be covered by sequential implementations and composed with others, transparently to the user. We present additional details on combining these operators in the discussion section.

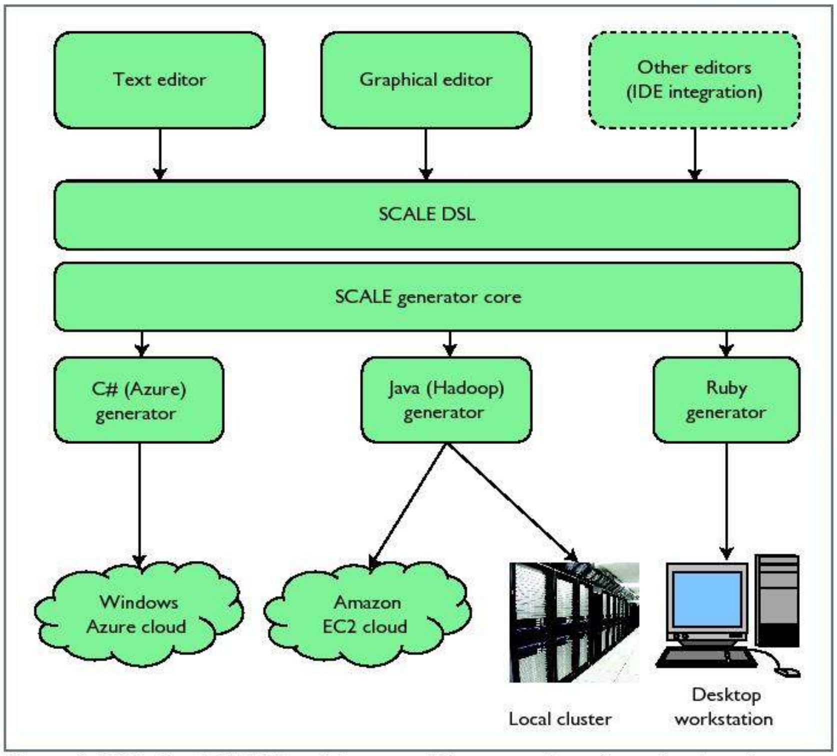

Figure 1. High-level SCALE architecture. Users can formulate the domain-specific language (DSL) script in many ways; the parser and generator components then use it to generate platform-specific code.

Architecture and Tools

The SCALE architecture is centered on the DSL and follows the best practice principles for DSL-based code generation.[2] Scientists can create the DSL script via many methods, such as using a graphical editor or a text editor. Once created, the DSL script is processed by a common parser generator infrastructure that creates a semantic model of the workflow. The generator then passes this model through the required platform-specific generators to create the code. Figure 1 illustrates SCALE's high-level architecture.

To reduce the local tooling requirements to a minimum, we built the SCALE composer as a Web-based tool. Scientists can compose the workflow, download an appropriate implementation, and optionally deploy to a supported public cloud, all without having a specific local tool set. Figure 2 illustrates the SCALE composer's user interface, where users can compose their code by dragging and dropping icons representing operators. The figure includes the partial visual representation of the workflow discussed earlier as an example.

The current generator supports the following target platforms:

- *Local (desktop) application.* The Ruby application can be used in a desktop environment and is suitable for small datasets. Users can download this code, which requires a local Ruby environment.
- *Windows Azure application.* The C# application uses the Microsoft

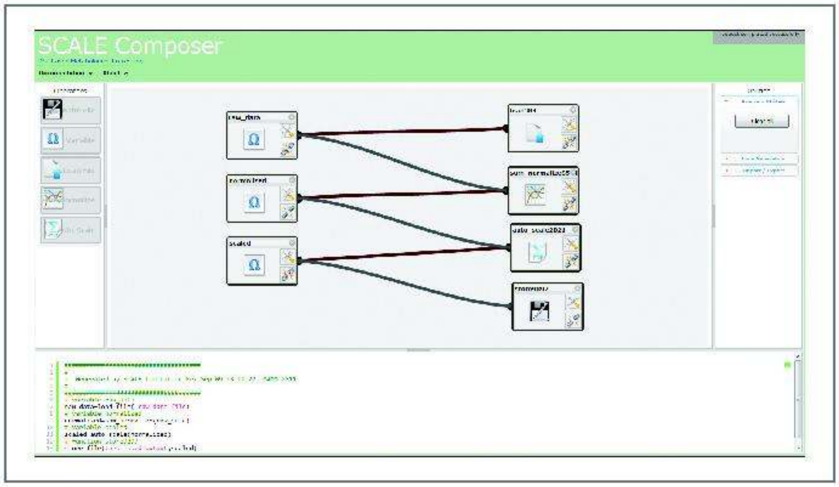

Figure 2. The SCALE composer. This screenshot illustrates the sum normalization and auto scaling operations.

Research Daytona runtime for MapReduce (http://research.microsoft.com/en-us/projects/daytona/default.aspx), based on Windows Azure. Users can deploy this application to an Azure cloud account. The generated code is provided as a Visual Studio project.

- *Apache Hadoop application.* This Java application uses the Apache Hadoop runtime and can be deployed in either a local cluster or the Amazon Elastic Compute (EC2) cloud. The tool provides a convenient wrapper to directly deploy the application to Amazon EC2, using the Amazon's elastic MapReduce system.

A version of the SCALE composer is available publicly at http://metabolink.knoesis.org/SCALE.

Opportunities for Domain Scientists

The SCALE system provides many benefits to a metabolomics scientist and e-Scientists with similar requirements.

Interoperability of Workflow Specifications

The workflow specification is now in a platform-agnostic, self-documented form. This makes collaborating with other scientists extremely easy, even when they aren't accustomed to the specification syntax or language. Scientists can use the specification to generate an appropriate implementation and reproduce experiments with ease.

Faster Time to Process

The scientists themselves, rather than a dedicated IT staff, can now form the workflows, allowing for faster creation. The composer supports direct deployments in some cases, which further reduces the effort required. A detailed comparison, highlighting the reduction in effort, is present online in the SCALE Toolkit experiment section at http://metabolink.knoesis.org.

Ability to Scale

The presence of multiple implementation options gives scientists the opportunity to test a small dataset in the desktop and seamlessly migrate to a cluster or cloud when a larger scale is needed. If using a cloud, scientists can go for an appropriately sized cluster, depending on the urgency and the expense.

Cost-Effective

SCALE achieves cost-effectiveness by requiring less staff to form and deploy workflows and providing flexible deployment, such that processes can now be run on a cloud or an available cluster. Scientists can employ clouds on a pay-per-use basis, without any upfront costs, saving on expensive hardware acquisitions.

Provenance Integration

Provenance has become an important aspect of scientific data processing, yet managing provenance data is a cumbersome task. The generated programs can easily have extensions to record the necessary provenance data (the exact nature of the statistical processing task, when, where, by whom, and so on), automating the bulk of the provenance-management process.

Discussion

During the implementation of the SCALE generators, we observed that some operators, such as OPLS, aren't naturally parallel algorithms. These algorithms are difficult to convert to parallel implementations and, even if converted, result in an inefficient implementation. Consequently, we focused on operations that are more straightforward to convert to parallel implementations for our initial implementation. Although we're implementing some other common operators as well as investigating the possibility of redesigning some commonly used advanced operators such as OPLS, the nonparallel implementations can still be used with the parallel ones, transparently to the user. For example, the OPLS implementation is still sequential, yet users can incorporate it with other operators even when other implementations aren't sequential. With OPLS, the input is much smaller than the original raw data and manageable within a single process. However, OPLS is iterative, and being able to convert it into a parallel implementation will definitely improve the overall process's speed and efficiency.

Metabolomics researchers regularly tweak and modify statistical algorithms and even introduce new ones. We plan to introduce an extension capability to the language to let advanced users create extensive customizations. Extensions will enable scientists to customize existing algorithms and introduce new ones.

Scientific tooling has improved significantly over the past decade, increasing the amount of experimental data available to scientists. Unfortunately the processing and analysis methods for such large amounts of data have not seen a growth of similar scale.[4] Adopting recent advances in technology, such as cloud computing, for scientific data analysis has become a difficult choice due to vendor lock-in and the need of cross-disciplinary expertise spanning biology and informatics. There is a clear need for methodologies that can create scalable, cloud-capable, platform-agnostic programs.

The SCALE system demonstrates our approach's viability in the domain of metabolomics. Although it has ample room for improvement, SCALE clearly indicates that providing abstractions over a key set of operators significantly reduces the effort needed to create an e-Science data processing workflow and allows scientists to efficiently use available cloud computing resources in a platform-agnostic manner. We believe that such abstract driven methods are applicable to many other e-Science domains as well.

References

1. A. Manjunatha et al., "Identifying and Implementing the Underlying Operators for Nuclear Magnetic Resonance-Based Metabolomics Data Analysis," *Proc. 3rd Int'l Conf. Bioinformatics and Computational Biology*, Int'l Soc. Computers and Their Applications, 2011; http://knoesis.wright.edu/library/resource.php?id=1031.
2. M. Fowler and R. Parsons, *Domain-Specific Languages*, Addison-Wesley Professional, 2010.
3. D. Mahle et al., "A Generalized Model for Metabolomic Analyses: Application to Dose and Time-Dependent Toxicity," *Metabolomics*, vol. 7, no. 2, 2011, pp. 206–216; http://dx.doi.org/10.1007/s11306-010-0246-3.
4. T. Hey and A. Trefethen, "The Data Deluge: An e-Science Perspective," *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, G. Fox and T. Hey, eds., John Wiley & Sons, 2003; doi:10.1002/0470867167.ch36.

Ajith Ranabahu is a PhD candidate in computer science in the Kno.e.sis center at Wright State University. His primary research is focused on application and data portability in cloud computing. Contact him at ajith@knoesis.org.

Paul Anderson is an assistant professor and the director of the Bioinformatics Research Group at the College of Charleston. His research interests include informatics, computational science, and e-Science. Contact him at andersonpe2@cofc.edu; http://birg.cs.cofc.edu.

Amit Sheth is the director of the Ohio Center of Excellence in Knowledge-enabled Computing (Kno.e.sis) at Wright State University. His focus is on developing semantic approaches and background knowledge to process, integrate, analyze, understand, and make actionable a wide variety of sources, including scientific experiments and literature, social media, and sensors. Sheth has a PhD in computer and information science from Ohio State University. He's a fellow of IEEE and a LexisNexis Ohio Eminent Scholar. Contact him at amit@knoesis.org; http://knoesis.org/amit.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*