

Facilitating Bioinformatic Research with Mobile Cloud

Jinhui Yao^{1,2}, Jingyu Zhang², Shiping Chen¹, Chen Wang¹, David Levy²

¹*Information Engineering Laboratory, CSIRO ICT Centre, Australia*
 {Firstname.Lastname}@csiro.au

²*School of Electrical and Information Engineering, University of Sydney*
 {jinhui, jy Zhang, dlevy}@ee.usyd.edu.au

Abstract—In this paper, we propose a concept of mobile-cloud by combining mobile and cloud together in a bioinformatic research application scenario. A mobile-cloud framework is developed, which facilitates the use of mobile devices to manipulate and interact with the scientific workflows running in the Cloud. In this framework, an independent trusted accountability service is used to provide data provenance and enforce compliance among the participants of a bioinformatic workflow. We have implemented a prototype which allows the bioinformatic workflow design and participation using mobile devices. We prove the concept of mobile-cloud with the prototype and conducted performance evaluation for the significant points of the bioinformatic workflow.

Keywords—cloud computing; accountability; service oriented architecture; mobile cloud.

I. INTRODUCTION

The emergence of computing resource provisioning known as the Cloud has revolutionized the modern day computing. It has provided a cheap and yet reliable outsourcing model for whoever with huge needs for computing resources. Given the fact that many scientific breakthroughs need to be powered by advanced computing capabilities that help researchers manipulate and explore massive datasets [1], Cloud offers the promise of “democratizing” research as a single researcher or small team can have access to the same large-scale compute resources as large, well-funded research organizations without the need to invest in purchasing or hosting their own physical infrastructure.

On the other hand, the concept of Service Oriented Architecture (SOA) allows flexible and dynamic collaborations among different service providers. A service can either directly be used for its mere function or be composed with other services to form new value-added workflows [2]. Through SOA, scientific workflows can be used to bring together these various scientific computing tools and compute resources offered as services in the Cloud to answer complex research questions. Workflows describe the relationship of the individual computational components and their input and output data in a declarative way. In astronomy, scientists are using workflows to generate science-grade mosaics of the sky [3], to examine the structure of galaxies [4]. In bioinformatics, researchers are using workflows to understand the underpinnings of complex diseases [5].

In scientific workflows, certain critical steps need the participation of respective research personnel or experts. For

example, how the workflow should be designed and which scientific tools need to be involved must be decided by the expert in the area. And some complex patterns generated from the experiments need to be visually inspected by the scientists who will based on their domain knowledge and experience determine the next a few steps for further analysis. In this regard, it is highly desirable that scientists can have easy access to the services in the Cloud so that they can design and participate in the workflows efficiently.

To address the above need, with the impressive advanced in the technology, we believe using mobile devices can be an ideal solution. The processes in a workflow can be thoroughly integrated with portable devices. All activities are decided and monitored on time from the way that fit the human environment instead of forcing users to passively accept the computing results from cloud service. In this paper, we propose a novel design which facilitates the use of mobile devices to manipulate and interact with the scientific workflows running in the Cloud. In our system, the users can choose the services in the Cloud to form the workflows via their mobile devices, and each mobile device can serve as one service node to be involved in the workflows designed. The contribution of this paper is two fold, we first elaborate our framework enabling mobile devices to compose and participate in the workflows running in the Cloud. Then, we further propose our approach to incorporate accountability into the system in order to enforce compliance and provide data provenance.

II. THE APPLICATION SCENARIO

In the area of gene research, the recent development of the microarray technology [6] have led to rapid increase in the variety of available data and analytical tools. Some recent surveys published in Nucleic Acids Research describes 1037 databases [7] and over 1200 tools [8]. The analysis of microarray data commonly requires the biologist to query various online databases and perform a set of analysis using both local and online tools.

To illustrate with an example, here we explain the research study of the genetic cause of colorectal cancer, i.e., identify the genetic variation in human DNA that makes people susceptible to colorectal cancer. By studying the functions of the genes involved, biologists can have a better understanding of the cancer and find possible cure. The first step in this study is to perform experiments on mice, which share more than 90% DNA with human. Microarray experiments are performed on

both cancerous and healthy mouse colon tissues [6]. By comparing the results from mice with and without colorectal cancer, biologists can identify candidate genes that may cause the cancer. Further analysis—such as searching for the functions known to these genes—are commonly performed to examine whether and how the candidate genes relate to the colorectal cancer. The followings are some standard analysis that are required for the study of microarray experiment results:

Quality Control. The raw microarray result data are processed, visualized and inspected by an expert, who can identify errors and discard the experiment.

Normalization. Microarray results from different samples need to be normalized before any meaningful comparison can be conducted.

Gene Differentiation. By contrasting the results from cancerous and healthy tissues, differentially expressed genes—genes that are active in cancer but not healthy tissue or vice versa—are identified.

Gene Study. Most differentially expressed genes are further studied to understand the biological foundation of the disease. Many resources are available to study these genes, such as the gene sequence, pathway database (e.g., KEGG), and gene function ontology. Experts need to be involved to make good decision as which study to conduct and which database to use.

We can see that the four standard analysis procedures we listed above not only can be extremely computing intensive but also require some decision making from the research scientists or experts at certain critical steps (e.g., quality control). It easily follows that, a viable approach to conduct such researches must utilize certain computing platform that has enormous computing capacity, yet research scientists can easily interact with the platform and the computing process conducted. This is essentially the reason for which we promote the “Mobile Cloud” - a composition of the Cloud, and the mobile devices - to be a suitable paradigm for complicated bioinformatics researches.

III. A MOBILE-CLOUD SYSTEM FOR BIOINFORMATIC RESEARCH

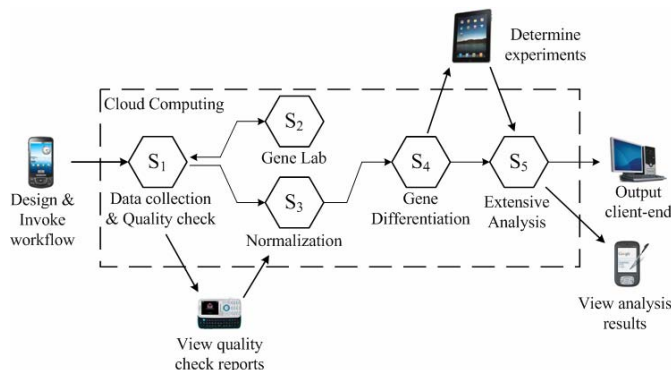


Figure 1. Overview of the proposed system

As we have established in previous sections, we propose to compose the Cloud and the mobile devices to conduct complex bioinformatics researches. The bioinformatics research scenario

we chose is the study for the cause of colorectal cancer. Fig. 1 shows our proposed system with this research scenario.

In the Cloud, different computing intensive gene research tools are deployed by different research bodies and provided as services. Outside the Cloud, research scientists or gene analysts locate the desired services in the Cloud, and use them to compose a workflow for studying the cancer. The workflow starts with retrieving raw microarray data from the nominated “Gene Lab”, after going through a sequence of processing, produce the final output to send to the “Output client-end”. Multiple research personnel may be involved in the workflow, they participate in the workflow by using mobile devices to invoke or receive output from the services.

Our argument for using mobile devices to design and participate in the workflows is intuitive. As mentioned, in the workflow there are “critical steps” that require decision making by experts in the respective area, in order to continue the process. For example, after the quality check, an important decision needs to be made about whether the quality of the raw data suffices the requirements of the experiment. The experiment should be paused before the expert in charge has reviewed the quality check reports and confirmed the usability of the raw data. Therefore, mobile devices are indeed ideal for this task for its outstanding mobility compared to desktop computers or even laptop computers, i.e., one can freely use his mobile devices while waiting in a queue, on a bus, or even walking. Further, given the recent impressive advances in the mobile technology, the computing capability of mobile devices - however limited compared to desktops or laptops - is more than enough to run basic UI or display data sets and processing reports. Therefore, we believe mobile devices such as smart phones or tablet computers are indeed ideal to be used as light client-end to drive the heavy bioinformatic research workflows in the Cloud.

To enable mobile devices to construct and participate in the workflows running the Cloud, we have developed the Mobile Cloud middleware layer (MC-layer) to facilitate these. Fig. 2 provides an overview of the architecture, which consists of a user interface (residing on mobile devices), a Cloud environment containing various services and a middleware layer consists of three function units. Their respective functionalities are summarized as follows:

- **Cloud Environment** provides various services deployed by respective providers. The services have registered their access end point with the MC-layer.
- **Service Repository** stores the information about the services in the Cloud that has registered with it. Once a search request is received, it will find the best service or workflow that satisfy the functional and non-functional requirements specified.
- **Service Composition** is responsible of composing individual services into workflows.
- **Service Execution** conducts two jobs: (a) orchestrating workflows; (b) invoking Web services.
- **User Interface** allows users to register, design workflows and participate in a running workflow.

For mobile devices to construct workflows, they first need to send a search request to the Service Repository in order to get a list of the services/workflows they are looking for. A convenient UI has been implemented on the mobile devices to allow the users to easily design the workflows using the services listed by the Service Repository (the UI will be elaborated in the evaluations). Once the workflow have been designed, a representative XML based description script is generated to be submitted to the Service Composition unit. The Service Composition unit thus according to the script, composes the services to form the desired workflows. The services can be composed in two ways: i) centrally composed, where the MC-layer invokes the services in the sequence designed by the user; and ii) remotely orchestrated, where certain orchestration scripts such as BPEL will be generated and distributed to all the services involved for deployment.

In our system, the workflow designed by the users is an abstract workflow, that is, the users only need to specify the type of service needed, and the MC-layer will search its service repository and select the best suited ones according to the user’s specifications. Table 1 gives a sample of the workflow description script. As it is developed based on the BPEL, “sequences” and “flows” are used to specify serial and parallel composition, and “Actions” are used define the invocation operations. The sample describes the first half of the gene analysis workflow in Fig. 1. In some actions, the endpoint is set to be “OPTIMAL”. This is to tell the Service Composition unit to choose the best suited services.

TABLE I. SAMPLE WORKFLOW DESCRIPTION SCRIPT

```

<sequence name="main">
  <Action operation="start" invoker="client" endpoint="QualityCheck"
  type="send&forget".../>
  <Action operation="fetchGene" invoker="QualityCheck"
  endpoint="GeneLab" type="send&receive".../>
  <flow>
    <Action operation="sendForApproval" invoker="QualityCheck"
    endpoint="OPTIMAL" type="send&forget".../>
    <Action operation="normalization" invoker="QualityCheck"
    endpoint="OPTIMAL" type="send&forget".../>
  </flow>
  ...
</sequence>
    
```

As we have established in our system design, mobile devices will be involved in the workflows as web services. To facilitate this, we created a customized web service engine to run on the mobile devices. Using this engine, mobile devices can both send and receive service requests, as well as interpreting the workflow description scripts delivered by the MC-layer. Once a user has designed and submitted a workflow, the workflow description script will be forwarded to the research personnel that are involved. The mobile devices they are using will interpret the workflow script and save the workflow logic. When a service request is received during the execution of the workflow, the UI will allow the user to view the content (e.g., quality check reports) and provide the list of the services that the user should send output request to according to the workflow logic (e.g., normalization services). For the technical details of the MC-layer, please refer to our

previous publications about the Web Service Management System (WSMS) [12].

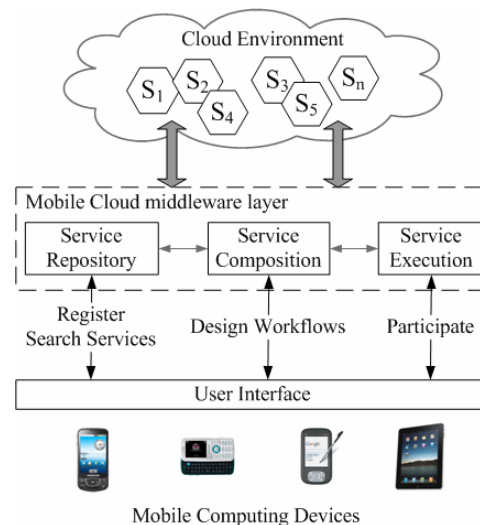


Figure 2. Overview of Mobile Cloud architecture

IV. ACCOUNTABILITY FOR COMPLIANCE AND PROVENANCE

The workflows in the Cloud are constructed using services provided by different parties who barely know each other. The correctness of the resultant workflow relies on the individual correctness of all participators. That is, if the service is compliant to the pre-defined workflow logic, or Service Level Agreement (SLA). The scientific integrity of the gene analysis results will be highly questionable if the services involved can act willy-nilly and get away with processing errors.

On the other hand, for scientific experiments not only the resultant data are considered, the steps of how these data are derived along the process can also be very valuable. It has been widely realized that data provenance plays an important role in the scientific researches [13]. It follows that, a mechanism is clearly needed to preserve the intermediate data forms generated by different services during the execution of the workflow, for compliance monitoring and provenance of the analysis results. We regard this mechanism critical for the viability of the paradigm we have proposed. In this section, we illustrate our design to incorporate accountability into the “Mobile Cloud” to address these issues.

Accountability can be interpreted as the ability to have an entity account for its behaviors to some authorities [9]. This is achieved by binding each activity conducted to the identity of its actor with proper evidence [10]. Such binding should be achieved under the circumstance that all actors within the system are semi-trusted. That is, each identified actor may lie according to their own interest. Therefore, accountability should entail a certain level of stringency in order to maintain a system's trustworthiness. Below, we identify several desirable properties of a fully accountable system:

- **Verifiable:** The correctness of the conducted process can be verified according to the actions and their bindings recorded.
- **Non-repudiable:** Actions are bound to the actors through evidence, and this binding is provable and undeniable.

● **Tamper-evident:** Any attempt to corrupt to recorded evidence inevitably involves the high risk of being detected.

We illustrate our proposed approach in Fig. 3. In our approach, accountability can be incorporated into activity-based workflow by requiring the entity conducting the process to log non-disputable evidence about the activities in a separate entity. In the figure, after incorporating accountability into an ordinary process, entity A is now required to perform logging operations before and after conducting the activity in its process. The evidence is logged in a separate entity - entity B - so that entity A cannot access the logged evidence. The evidence needed to be logged should contain enough information to describe the conducting activity. In our simple example, which is intuitive enough, the evidence should include the states of the factors concerning the start of the activity (e.g., the input variables) and the factors concerning its completion (e.g., the output value).

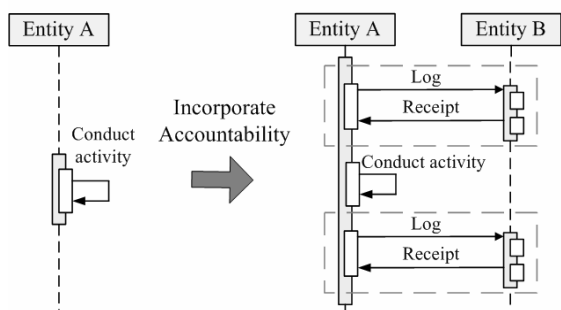


Figure 3. Example of incorporating accountability into process

The logging operations require the employment of PKI in all involved service entities. Each of them has its own associated public-private key pair issued by certificated authorities. The logging operations are as follows:

1. The logger (entity A) signs the evidence (E) by its private key (K_A) to create a digital signature of the evidence (S_A).
2. The evidence and its signature are then logged in a separate entity (entity B).
3. When received, entity B creates a receipt by signing entity A's signature with entity B's private key (K_B).
4. Lastly, the receipt (S_B) is sent back to the logger (entity A) in the reply.

Assuming the digital signature is un-forgable, the signed evidence in entity B can be used to verify entity A's compliance; and yet any corruption or deletion applied to the evidence will be discovered using the receipt received by entity A. Under the circumstance that neither of the service entities is trusted; and assume they will not conspire to cheat, this structure manages to ensure the proper preservation of evidence associated with the process conducted.

To have the separate entity B to preserve the evidence, we propose to have special service nodes, dedicated to provide accountability to all underlying services involved in the workflow. Those special nodes are referred to as the

accountability service (AS) nodes. Fig. 4 shows the structure. All the mobile devices, service nodes in the Cloud as well as local computing nodes that are involved in the workflow, register with AS nodes and submit evidence during the execution of the workflow. The implementation details of the incorporation of accountability have been elaborated in our previous work [11].

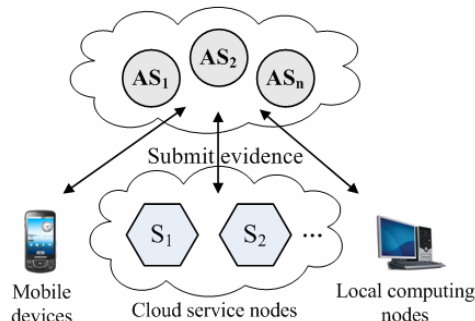


Figure 4. Accountability Service (AS) for compliance and provenance

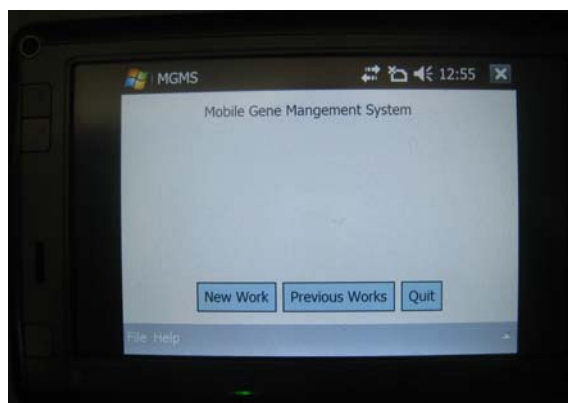
Here the evidence can be any intermediate gene analysis data generated by the tools in the Cloud, or the decisions made by research personnel participated. With the evidence data logged, the core functionalities provided by the AS nodes are:

- Compliance verification. Through the analysis of the evidence data, the correctness of the behaviors of the underlying services is continuously validated.
- Data provenance. The evidence recorded capture the evolution path of the data as well as the entities responsible for each step.
- Workflow status monitoring. A global view over the workflow is maintained by the AS nodes. Such information can be used to assist the functioning of the MC-layer and the underlying services.

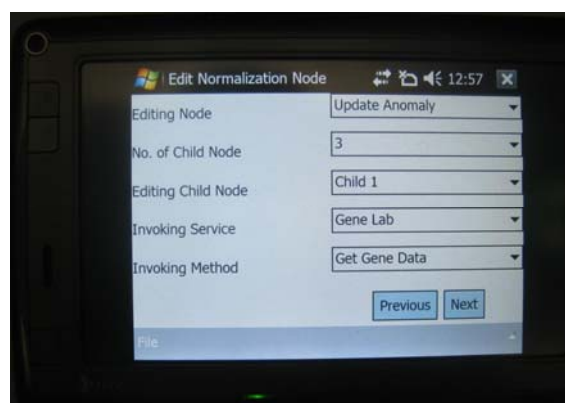
The AS nodes can either be provided by the Cloud, or by other third parties as long as they receive no benefit whether the underlying services are being compliant or non-compliant. They play a neutral role. Note that it is undesirable to build the accountability mechanisms into the MC-layer as it is the entity which designs and orchestrates the workflow and is also subject to errors. Using AS nodes provided by unrelated third parties offers a higher level of honesty and stringency.

V. EVALUATIONS

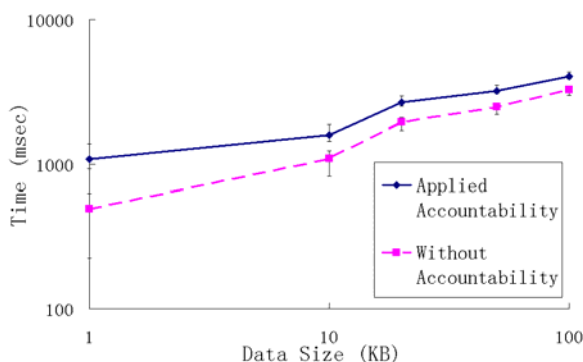
We prototyped a demonstration system to showcase our mobile-cloud concept. Our system consists of four parts: i) a client UI deployed in the mobile device; ii) an MC-layer to search and compose services; iii) a number of demonstrating service nodes in Amazon EC2; and iv) an accountability service. In this section, we will first elaborate the implementation of the client UI; then we compare the communication latency when the accountability service is involved and uninvolved; finally we shows some processing latency when a real gene database (KEGG) is involved in a workflow running in the Cloud.



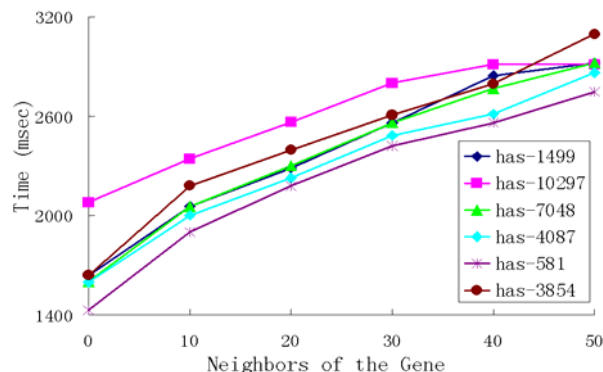
(a) Main User Interface of MGMS



(b) Design a workflow in MGMS



(c) Communication cost with AS and without AS in our system



(d) Performance of invoking KEGG service from the workflow engine

Figure 5. Mobile User Interface and Performance Evaluation

The UI on mobile device is developed using Java platform, micro edition (J2ME). The mobile web service feature is deployed and runs on a HTC 9500 mobile phone, which is running on IBM Websphere Everyplace Micro Environment that supports a connected device configuration (CDC1.1). Figure 5 (a) and (b) show two screen shots of the Mobile Gene Management System (MGMS) - a scientific workflows design and surveillance tools. A user can define or edit a scientific process from the “New Work” button or “Previous Work” button as shown in Figure 5 (a). Then, the user can select into process items and specify their detail information as shown in Figure 5 (b). System users define the steps from four aspects, what services carry out these tasks; the number of child nodes; which methods/services are invoked; and what are the inputs and outputs of each step. Finally, an abstract workflow in BPEL will be generated and uploaded to the WSMS in Cloud, which will instantiate the abstract workflow by filling up the endpoints in the BPEL with the best concrete services URLs.

Figure 5 (c) shows the interaction latency between mobile device, cloud nodes and the AS, with the average value being 492.7msec at 1kB and 3251.2msec after the communication size is increased to 100kB. According to our processes, mobile and cloud nodes need to communicate with the AS so the average value being 660.5msec for the whole system regardless the underlying operation work load. From the curve of this figure, we see the percentage that cost on AS is

decreased from 54% to 19% with the communication size increasing from 1kB to 100kB.

To evaluate the performance of gene retrieving from gene bank services, we selected 6 example genes which are the genetic causes of colorectal cancer and retrieve their genetic neighbors from KEGG disease Database [21]. We test the response time from 0 neighbors to 50 neighbors. As shown in Figure 5 (d), it is clear that the latency is slowly increasing with changing the number of neighbors. The has-581 continually kept the best performance at all stages from the 1427msec for retrieving 0 gene neighbor to 2746.8msec for getting 50 neighbors. However, has-10297 spent 2078msec to search 0 neighbors and it cost 2912.6msec for finding 50 neighbors.

VI. RELATED WORK

Mobile computing provides a luggable computation model for users. Its portability makes it very ideal for many application scenarios. To extend its limited computing power, research communities have proposed novel designs to leverage the Cloud. Huerta-Canepa and Lee [22] proposed a virtual cloud system, Zhang et al. [23] detailed a distributed computing platform using mobile phones. They improve the capacities of mobile phones in the purpose of storage and computation. Works like [24-26] presented some compu-

tation offloading methods that move some parts of the applications to run on the Cloud. Executing parts of application remotely can save battery lifetimes and significantly extend computing resources. However, these solutions do not support platform-independent cooperative interaction over an open network. In addition, after moving some parts of applications from stand-alone handheld devices to the cloud, several issues need to be considered in advance such as privacy, trustworthy or provenance.

The importance of provenance for scientific workflows has been widely acknowledged by various research communities. Many approaches have been proposed to record the derivations of the data during the scientific process. Approaches like [14][15] allow the designer to capture the intermediate data forms generated by the experiments at different granularities. In our work, we introduced the concept of accountability which not only provides data provenance but can enforce compliance among the service providers. Compliance assurance has been studied decently in recent years, some remarkable works include [17] [18] [19] [20]. Our work differs from them at the point that we consider a more hostile environment where all service entities are expected to behave in any possible manner and deceive for their own benefit. Cryptographic techniques are deployed in our system to ensure the evidence are undeniable.

VII. CONCLUSION

In this paper, we have proposed a novel design to enable mobile devices to design and participate in the scientific workflows running in the Cloud. The scientific researchers can use mobile devices to sketch an abstract workflow design to be submitted to the mobile cloud middleware layer, which will choose and compose the optimal services according to the designer's requirements. On top of that, we further incorporated accountability mechanisms not only to provide data provenance during the process but also enforce compliance among all the service providers involved. Our testing data indicate that the cost of incorporating accountability is acceptable and becomes negligible when the transmission data become large.

REFERENCES

- [1] W. Lu, J. Jackson, and R. Barga. AzureBlast: A Case Study of Developing Science. In proc. Workshop on Scientific Cloud Computing, pp. 413-420, 2010.
- [2] O. Moser, F. Rosenberg, and S. Dustdar. Non-Intrusive monitoring and service adaptation for WS-BPEL. In proc. international conference on World Wide Web, pp. 815-824, 2008.
- [3] Montage. <http://montage.ipac.caltech.edu>. Last accessed 8th Sep 2011.
- [4] I. Taylor, M. Shields, I. Wang, and R. Philp. Distributed P2P computing within Triana: A galaxy visualization test case. In proc. IEEE International Parallel and Distributed Processings Symposium, 2003.
- [5] T. Oinn, P. Li, D.B. Kell, C. Goble, A. Goderis, M. Greenwood, D. Hull, R. Stevens, D. Turi, and J. Zhao. Taverna/myGrid: Aligning a workflow system with the life sciences community. In *Workflows in e-Science*, Springer, 2006.
- [6] M. Schena, D. Shalon, R.W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467-470, October 1995.
- [7] M. Y. Galperin. The molecular biology database collection: 2008 update. *Nucleic Acids Research*, pages gkml037+, November 2007.
- [8] M. D. Brazas, J. A. Fox, T. Brown, S. McMillan, and B. F. F. Ouellette. Keeping pace with the data: 2008 update on the bioinformatics links directory. *Nucleic acids research*, 36, July 2008.
- [9] R. Mulgan. Accountability: An ever-expanding concept? In: Public Administration, pp. 555-573, 2000.
- [10] A. R. Yumerefendi, and J. S. Chase. Trust but verify: accountability for network services. In proc. ACM SIGOPS European workshop, article No. 37, 2004.
- [11] J. Yao, S. Chen, C. Wang, D. Levy, and J. Zic. Accountability as a service for the cloud, in proc. IEEE International Conference on Services Computing, pp. 81-90, 2010.
- [12] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed. Deploying and managing web services: issues, solutions, and directions. *VLDB J.*, 17(3):537-572, 2008.
- [13] Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science, in trans. ACM SIGMOD Record, volume 34, issue 3, pp. 31-36, September 2005.
- [14] I. T. Foster, J.-S. Vöckler, M. Wilde, and Y. Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation, in *SSDBM*, 2002.
- [15] J. Zhao, C. A. Goble, R. Stevens, and S. Bechhofer. Semantically Linking and Browsing Provenance Logs for Escience, in *ICSNW*, 2004.
- [16] J. Myers, C. Pancerella, C. Lansing, K. Schuchardt, and B. Didier. Multi-Scale Science, Supporting Emerging Practice with Semantically Derived Provenance, in ISWC workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, 2003.
- [17] E. Mulo, S. Dustdar, and U. Zdun. Monitoring Web Service Event Trails for Business Compliance. In Proc. International Conference on Service-Oriented Computing and Applications , pp. 1-8, 2009.
- [18] M. Huang, L. Peterson, and A. Bavier. PlanetFlow:maintaining accountability for network services. In Proc. ACM SIGOPS Operating Systems Review, pp. 89-94, 2006.
- [19] Y. Zhang, K. Lin, and J. Y. J. Hsu. Accountability monitoring and reasoning in service-oriented architectures. In Trans. Service Oriented Computing and Applications, Volume 1, Number 1, pp. 35-50, 2007.
- [20] A. C. Squicciarini, W. Lee, B. Thuraisingham, and E. Bertino. End-to-end accountability in grid computing systems for coalition information sharing. In Proc. Workshop on Cyber Security and Information Intelligence Research , 2008.
- [21] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa. KEGG for representation and analysis of molecular networks involving diseases and drugs. In trans. *Nucleic Acids Research*, volume 38, Database issue, pp. 355-360, 2010.
- [22] G. Huerta-Canepa and D. Lee. A virtual cloud computing provider for mobile devices. presented at the Proceedings of the 1st ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond, San Francisco, California, pp. 61-65, 2010.
- [23] J. Zhang, David Levy, Shiping Chen, and John Zic. mBOSS+: A Mobile Web Services Framework. in Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific, pp. 91-96, 2010.
- [24] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso. Calling the cloud: enabling mobile phones as interfaces to cloud applications. the 10th ACM/IFIP/USENIX International Conference on Middleware, pp. 83-102, May,2009
- [25] K. Kumar and Y. Lu. Cloud Computing for Mobile Users. *Computer*, vol. 18, issue 99, pp. 51-56, 2010.
- [26] R. Kemp, N. Palmer, T. Kielmann, and H. Bal. Cuckoo: a Computation Offloading Framework for Smartphones. In *MobiCASE '10: Proceedings of The Second International Conference on Mobile Computing, Applications, and Services*, pp. 62-81, 2010.