GeoSquare: A cloud-enabled geospatial information resources (GIRs) interoperate infrastructure for cooperation and sharing

Kai Hu[1], Huayi Wu[1], Zhipeng Gui[2], Lan You[1], Ping Shen[1], Shuang Gao[1], Jie Zheng[1], Xiaoqiang Cheng[1]

[1] The State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China

[2] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China,

[3] Faculty of Computer Science and Information Engineering, Hubei University, Wuhan, China

1. Introduction

With the advancement of sensors and information technologies, a large amount of geographical information resources (GIRs), including geospatial data, algorithms, application and models, have been available on the internet for public use. Due to the heterogeneity and complexity of the abundant resources, the interoperation of the geospatial resources became a crucial challenge to the geo-computation(Gong et al., 2012). A lot of platforms aimed at sharing and collaboration of the GIRs were built to tackle the problems. By integrating registry and sharing functions of the geo-data and geo-service models, they were able to utilize the resources intuitively and offered a common communication platform for collaborative operations between researchers and teachers.

However, because of the GIRs' large-volume characteristic and the frequent user operations, the performance and scalability issues of the platforms become serious challenges. Secondly, under the normal architectures, the service and service chains and the to-be-processed data are often distributed in different internet places. The sharing become intolerable for the long delay of data transferring by the internet. Thirdly, when dealing with the input parameters of the geo-services, user will often find them too complex and uninterpretable. Unlike the normal web services, the geo-services are correlated to certain operation of the geo-data which is often different from the normal parameters. Based on that, we proposed a prototype system to tackle the problems.

2. Overview of our approach

With the mentioned problems taken into consideration, cloud-enabled computation came into our sight for its characteristics of large scale, virtualized, high reliability, commonality and scalability at a low price. In order to solve the mentioned problems, we promoted an infrastructure architectures named GeoSquare by combing with the widely used private cloud solution (Fig 1).

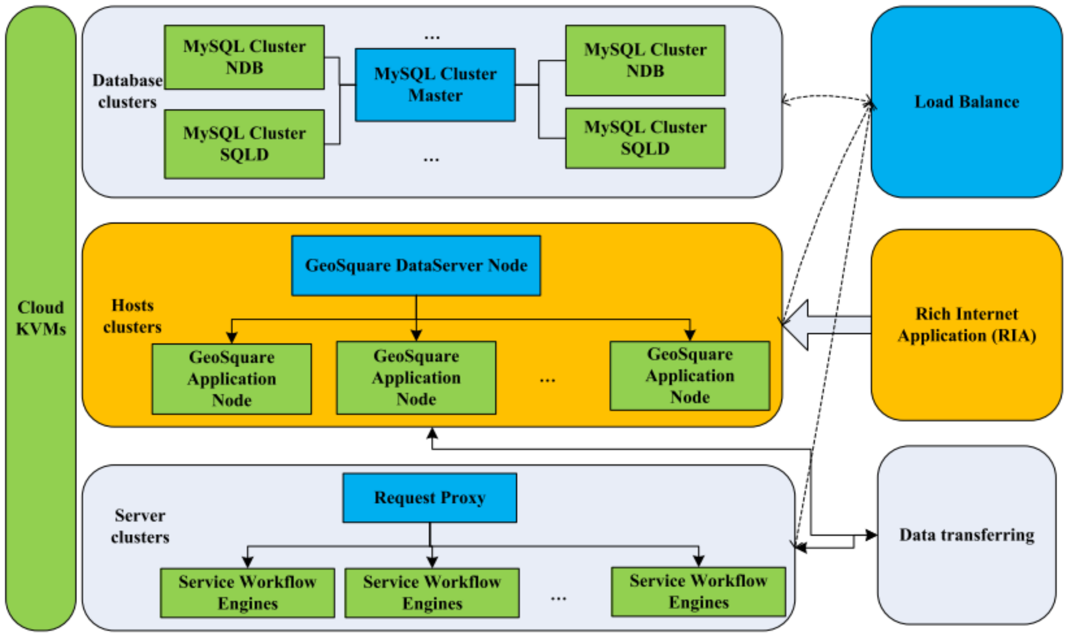

Figure 1. The architecture of cloud-enabled GeoSquare

In this architecture, we used the master-slave mode to solve the scalability problem and data transferring problems and extended the sharing forms in an attempt of a Rich Internet Application (RIA) to make the user interface more friendly. The promotions are described specifically as follows:

2.1 Performance improvement using cloud technology

To overcome the scalability problems of the normal sharing platform's, we chose the cloud solution to make load balance of large scales to host the geo data, geo services, geo service modules and the web host itself. By leveraging the convenient functions provided by the cloud, the architecture is designed as the one-data-node-multiple-application-node architectures which is convenient for scale up by setting the application-nodes' numbers. The MySQL-cluster is also used to promote the reliability of the metadata databases(Ronstrom and Thalmann, 2004). And the services workflow engines are expended in the load balance mechanisms in certain scales of server clusters. All of the MySQL-cluster nodes, the GeoSquare's application nodes and the service workflow engines are made into templates of the Kernel-based Virtual Machine (KVM) which provided by the cloud infrastructure(Habib, 2008). By using control center of the web interfaces of cloud management, the clusters' scale of applications, the databases, and the service workflow engines can be deliberately set to coper with different internet environment.

In this part, in order to testify our work efficiently proved the performance of the website, we conducted a stress testing experiment on our deployed websites. By using the software Apache Benchmark(Sheldon and Weissman, 2007), we conducted the web stress test as follows:

The cluster is constructed by five computer node and one of them is the proxy server. We use the Apache Benchmark with the parameter like Table1:

Single node test	Cluster balance test
ab –n 3000 –kc 50 http://singlenodeip/index.html;	ab –n 3000 –kc 50 http://clusternodeip/index.html;
ab –n 3000 –kc 100 http://singlenodeip/index.html;	ab –n 3000 –kc 100 http://clusternodeip/index.html;
ab –n 3000 –kc 150 http:// singlenodeip/index.html;	ab –n 3000 –kc 150 http://clusternodeip/index.html;
ab –n 3000 –kc 200 http://singlenodeip/index.html;	ab –n 3000 –kc 200 http://clusternodeip/index.html;
ab –n 3000 –kc 250 http://singlenodeip/index.html;	ab –n 3000 –kc 250 http://clusternodeip/index.html;

Table1. Request static file to test the performance of the website

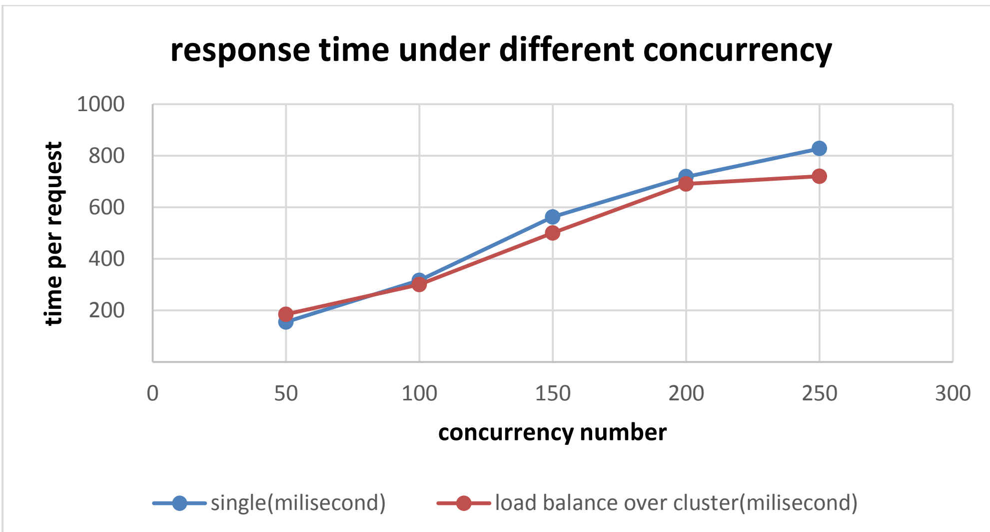

Figure 1.test on response time under high concurrency

With the same total amount of request of 3000, we changed the concurrency number, and we found that the website's performance is improved under different concurrency visit.

Single database(innodb) node test	Cluster database(ndb) balance test
ab –n 3000 -kc 50 http://singlenodeip/ webserive?request=GetPublic;	ab –n 3000 -kc 50 http://clusternodeip/ webserive?request=GetPublic;
ab –n 3000 -kc 100 http://singlenodeip/ webserive?request=GetPublic;	ab –n 3000 -kc 100 http://clusternodeip/ webserive?request=GetPublic;
ab –n 3000 –kc 150 http:// singlenodeip/ webserive?request=GetPublic;	ab –n 3000 -kc 150 http://clusternodeip/ webserive?request=GetPublic;
ab –n 3000 –kc 200 http://singlenodeip/ webserive?request=GetPublic;	ab –n 3000 -kc 200 http://clusternodeip/ webserive?request=GetPublic;
ab –n 3000 –kc 250 http://singlenodeip/ webserive?request=GetPublic;	ab –n 3000 -kc 250http://clusternodeip/ webserive?request=GetPublic;
ab –n 3000 –kc 300 http://singlenodeip/ webserive?request=GetPublic;	ab –n 3000 -kc 300 http://clusternodeip/ webserive?request=GetPublic;

Table 2. Request to interact with database to test the performance database

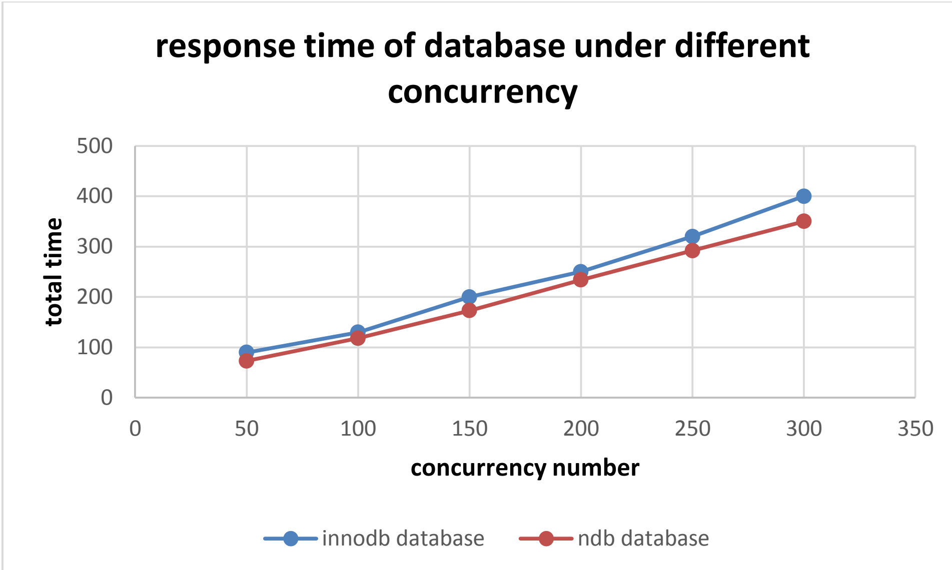

Figure 2.test on response time of interaction with database

To testify the performance of the databases, we also made 3000 requests to interact with the databases with different engines (single database using innodb and database cluster using ndbcluster) of the website from 50 to 300 concurrency and the results showed in the Figure 2 that, there is an main trend that total time of iteration with database was increasing with the development of the concurrency number. And under the different concurrency, the cluster database performance better than the single database with serval milliseconds ahead.

2.2 Data transferring in master-slave mode for service sharing

To deal with the time-consuming job of frequently data transferring in the service invocations, the data uploading and the data processing is departed in the design of the GeoSquare's distribution framework(Figure 3). The input parameters, which stand for the remote data, are altered to the results of querying the databases which links to the datacenter nodes' data. Then the service will invocate the inner data transferring from the data center to the services host nodes (Figure 4). And the data uploading function is provided to the users with the independent interfaces. Lastly, the records of the users operations (e.g. Service invocation, data uploading) is also recorded in the database and the result data of the operation can be used to share in the certain groups(Figure 5).

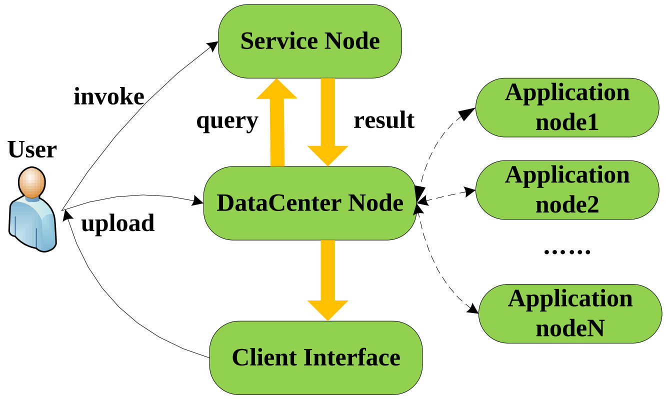

Figure 3.Data transferring in service invoking

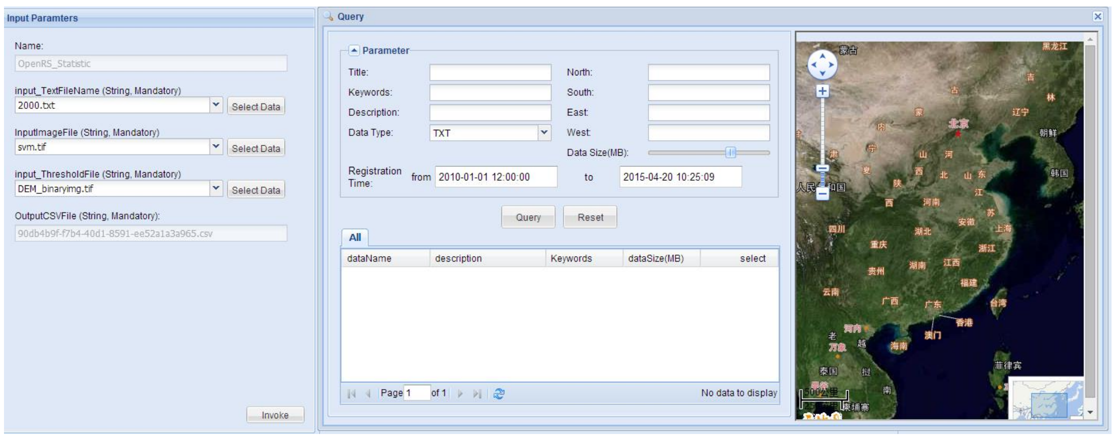

Figure 4. Interface of retrieving the data needed in a service

Figure 5. Sharing the results generated by invoked service to certain groups

2.3 Usability and HCI issues in promoting the sharing

As quoted, GIS were more likely to fail on human and organizational grounds than on technical ones(Goodchild, 2007). RIA is an good attempt for improving the usability of the web-based application and there are successful examples by adopt the technology(Huang et al., 2011).To make the services invocation more friendly, the application forms of the corresponding services and service chain models were tested in the attempt of several RIAs with the support of the scientific workflow technologies, the input mode was extended to the forms of interaction with the mash-up maps implemented by RIA technology instead of incomprehensible parameters(Figure 6).

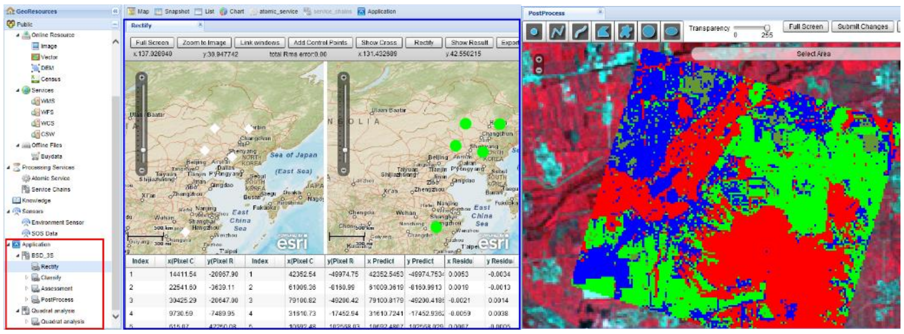

Figure 6. RIA interface used in sharing the algorithms in education

3. Conclusion

Though the above effort, GeoSquare is designed as a cloud computation enabled platform which is allowed for sharing the GIRs more efficiently within a larger user scales with a relatively more reasonable data transferring solution and the RIA is proved to be an efficient way to make the geo services more friendly. And as a whole, the sharing and collaboration capabilities of the platform is greatly improved.

4. Acknowledgements

This work was supported by Project supported by the National Natural Science Foundation of China (Grant No. 41371372).

5. References

GONG, J., WU, H., ZHANG, T., GUI, Z., LI, Z., YOU, L., SHEN, S., ZHENG, J., GENG, J., QI, K., YANG, W., LI, Z. & YU, J. 2012. Geospatial Service Web: towards integrated cyberinfrastructure for GIScience. *Geo-spatial Information Science,* 15**,** 73-84.

GOODCHILD, M. F. 2007. Citizens as sensors: the world of volunteered geography. *GeoJournal,* 69**,** 211-221.

HABIB, I. 2008. Virtualization with kvm. *Linux Journal,* 2008**,** 8.

HUANG, M., MAIDMENT, D. R. & TIAN, Y. 2011. Using SOA and RIAs for water data discovery and retrieval. *Environmental Modelling & Software,* 26**,** 1309-1324.

RONSTROM, M. & THALMANN, L. 2004. MySQL cluster architecture overview. *MySQL Technical White Paper*.

SHELDON, M. & WEISSMAN, G. V. B. Retrace: Collecting execution trace with virtual machine deterministic replay. Proceedings of the Third Annual Workshop on Modeling, Benchmarking and Simulation (MoBS 2007), 2007.