

# Enabling Remote Fault Diagnosis through Data-driven Grid Computing

Bing Tang, Li Zhang

*School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan, 411201, P.R.China*  
*b.tang@qq.com*

## Abstract

*Due to the complexity of modern manufacturing and mechanical equipments, it is difficult for equipment users or maintainers to accomplish fault diagnosis independently. In this paper, the development history of fault diagnosis technology is surveyed and investigated, especially remote fault diagnosis system based on Internet in detail. Then, a remote fault diagnosis system based on grid computing technology is proposed to enable collaborative resource sharing and problem solving among multiple equipment suppliers and equipment users by integrating all kinds of diagnostic resources. The architecture of this fault diagnosis system is presented, as well as the Client-Master-Worker computation model and the diagnostic workflow. Finally, a prototype system is implemented using the data-driven middleware-BitDew, in which multiple fault diagnosis grid services are integrated in a unified Web portal. The case study of data-driven fault data analysis is conducted in our prototype which has proved the effectiveness of the system.*

**Keywords:** *data-driven; grid computing; data-intensive computing; fault diagnosis*

## 1. Introduction

In modern manufacturing system, manufacturing equipments become more and more expensive, powerful, precise, intelligent, and complicated, which means that fault diagnosis also becomes more and more difficult, and is confronted with some new problems. The traditional fault diagnosis methods waste a great lot of resources, such as time and money, and the diagnostic results are hard to be achieved depending on equipment users or maintainers themselves or only one enterprise. It is also difficult for maintainers to get support quickly from maintenance service center that maybe located oversea, while the problem would be more serious in the global manufacturing mode. In this context, the most important and urgent thing is to find a new fault diagnosis approach to enhance the reliability and ability of manufacturing equipment fault detection, diagnosis and maintenance in the new manufacturing environment.

A good method to solve the above problem is to set up a remote intelligent fault diagnosis system based on network, in which equipment manufacturers, suppliers and users are integrated together to achieve remote collaborative fault diagnosis. To the best of our knowledge, there are many theoretical and practical research of fault diagnosis system based on network, such as fault diagnosis system for aircraft engine, ship engine, locomotive engine, large-scale electrical generator group, *etc.* In industry fields, Mitsubishi, Westinghouse Electric and Siemens Corporation have also researched remote services to solve fault diagnosis and maintenance for automation equipments in factory.

In general, fault data are collected by sensors or specific devices, which are various types of signals, such as vibration, noise, rotation speed, temperature, pressure, flow, *etc.* Through sensors, physical or chemical metrics are converted into electrical signals for store, and then sent to the diagnosis system for analysis and processing. As the rapid development of mechanical engineering, information and control technology, the

equipment always consists of several units or components, facing with some challenges as follows:

- (1) The collected data for fault diagnosis always change as the time, and storage infrastructure with large capacity is needed to store these data. The storage infrastructure may be distributed geographically, and multi-organizations may be involved in.
- (2) The diversity of fault and the indefinite model require advanced fault detection techniques, pattern matching techniques and data mining techniques. These techniques must be able to deal with Terabytes scale data, and must satisfy the requirement of response time.
- (3) Several collaborative participants are involved in the fault diagnosis system. A set of engineering computing tools are deployed to analysis the fault data, and an integrated fault isolation scheme must be obtained.

Meanwhile, we noticed that grid technology has emerged in the last decade, which brings a better solution for fault diagnosis of large-scale complicated manufacturing equipments. Grid technology realizes resource sharing and coordinated problem solving in dynamic, multi institutional virtual organizations, which is exactly matching the demand of fault diagnosis [1].

Currently some projects are developed to realize fault diagnosis in grid environment in the world. A famous project is the Distributed Aircraft Maintenance Environment (DAME) project which is supported by UK e-Science plan [2][3]. The purpose of this project is to set up a distributed aircraft maintenance environment, using Service-Oriented Architecture (SOA) technology, deploying diagnosis service in grid, and then realize the fault prediction and diagnosis of aircraft engine. Grid technology provides the fault diagnosis system with aircraft engine status data service, data store service, data mining service, engine modeling service, case-based reasoning (CBR) support and maintenance interface service[4]. Nevertheless, DAME considers about only the fault diagnosis of aircraft engine, not common equipments, which is the limitation of this project.

This paper presents a framework of grid-enabled remote fault diagnosis system to realize fault diagnosis for a variety of manufacturing equipments, which is an open and extendable system and can be widely used in manufacturing field. Multi-diagnostic resources are integrated and data-driven grid computing approach is used to analysis fault data. The system aims at establishing a common and unified fault diagnosis service platform for manufacturing equipments using grid technology.

The rest of this paper is organized as follows. Section 2 reviews the development history of fault diagnosis technology. Section 3 presents the architecture of proposed grid-enabled remote fault diagnosis system. The implementation of a prototype system is given in Section 4. Finally, Section 5 offers the concluding remarks.

## 2. Background and Related Work

### 2.1. The Early Stage of Fault Diagnosis Technology

Since the 1960s, when fault diagnosis is firstly researched in USA, by the effort of government, enterprises and scholars, fault diagnosis technology has been successfully used in many fields.

- (1) *The first generation technology*. Multi-users connection and centralize control are the main characteristics, and monitoring and diagnosis are realized in a stand-alone terminal. During this period, monitoring and diagnosis system is designed to fit a given machine or equipment, which is composed of a computer and one or more functional modules, and interaction and data processing are limited inside the system. Therefore, it is a closed system.

- (2) *The second generation technology.* Local area network (LAN) and distributed control are the main characteristics for this generation [5]. Because the maintenance and assistant components are distributed geographically, all components of large-scale electro-mechanical equipment are connected through industrial local area network. Hence, an industrial local area network system is designed to realize distributed monitoring and centralized operation, management and diagnosis. This is a relative open system, and monitoring and data processing are carried out inside the industrial local area network.

## 2.2. Remote Fault Diagnosis System based on Internet

After the 1990s, as the development of computer and information technology, monitoring and diagnosis technology have turned into a new stage, Internet-based *the third generation technology*. Remote monitoring and diagnosis based on Internet is the outcome of equipment diagnosis technology combining with computer network technology. In this diagnosis system, several computers are treated as servers, and state monitoring nodes are embedded into the important equipment to collect state data of equipment, and remote diagnosis and analysis center is established in scientific research institute, which provides remote technical support. Once the equipment works abnormally, state monitoring server sends out request to remote diagnosis server asking for online technical support, and sends out emails to experts asking for offline consultation at the same time.

Here is a brief description to the architecture of remote fault diagnosis system based on Internet. It follows a distributed hierarchical architecture which is composed of data gathering subsystem, local state monitoring and fault diagnostic subsystem, remote intelligent fault diagnostic subsystem (remote diagnostic center) and data communication network link [6].

- (1) *Data collection subsystem.* This subsystem accomplishes the task of gathering state data of equipment including static data and dynamic data. Industrial equipment often has sensors, actuators or PLC installed, which are used to acquire working status, sensor variable, position state and control parameter, *etc.* The collected fault data are used to analysis, estimate and fault localization.
- (2) *Local state monitoring and fault diagnosis subsystem.* This subsystem consists of information collection system, industrial computer and related software, which accomplishes the task of data management, ringing alarm, and real-time monitoring. The subsystem has a good GUI, presenting state of equipment and detailed fault diagnostic result in real-time, and also providing statistical analysis. The monitoring information is gathered, stored and processed, which is changed into compatible format for transmitting in networks after simply processed. Finally, related data and requests are sent out to remote fault diagnosis center server.
- (3) *Remote intelligent fault diagnosis subsystem.* Remote intelligent fault diagnosis subsystem is also called remote diagnosis center that is the core of fault diagnosis. Remote diagnosis center is often consisted of fault diagnosis experts from industry fields or institutes, in the form of human begin or remote diagnostic software. The main functions of this subsystem are listed as follows,
  - Receiving equipment state data from remote system;
  - Processing and analyzing monitoring data;
  - Integrating opinions of all experts;
  - Obtaining an integrated diagnostic result;
  - Providing approaches or advices to eliminate fault and maintain equipment;
  - Storing the current fault data and the case into the case-database.

Diagnosis center carries out the operation of decoding, and then corresponding fault diagnosis expert systems are called. Several experts are centralized together if necessary. A good diagnosis center depends on good expert system especially diagnostic software or program and diagnostic knowledge base. Diagnostic knowledge base has an open and extendable architecture. New knowledge is inserted into knowledge base under the management of administrator.

### 2.3. Remote Collaborative Fault Diagnosis System

Remote collaborative fault diagnosis system has the conspicuous characteristic of collaboration, which is called *the beyond third generation technology* in this paper. As we know, Internet bridges the gap between enterprises and scientific research institute in time and space. Academy fields hold theoretical technique, while enterprise fields hold valuable local experience and data. Hence, academies and enterprises could learn from others' strong points to offset one's weakness, and make extensive cooperation together.

The essence of collaboration is resource sharing. The eventual objective of collaboration is to realize remote fault diagnosis service sharing. The collaborative mode of remote fault diagnosis system improves economy efficiency, which also proposes several innovative characters, such as network, systematization, collaboration, loose-coupling, agility. In remote collaborative fault diagnosis system, the objects are abstracted as resources, and there are some categories of sharing [7], such as,

- (1) *Information resource sharing*. Information resource includes monitoring techniques, gathering techniques, equipment working status data, fault data, *etc.* Monitoring and gathering techniques can be adopted by study each other.
- (2) *Knowledge resource sharing*. Knowledge resource refers to expert knowledge, diagnostic knowledge, concepts, axioms, rules, *etc.* Web-based expert knowledge base is open and extendable, through Internet.
- (3) *Service resource sharing*. Service resource includes software algorithm or diagnostic algorithm, data analysis algorithm or software, signal processing algorithm or software. In a word, service resources are the algorithm or software to resolve problem.

Networks and interconnection are the foundation of sharing. In order to realize sharing of knowledge, theories and techniques, a common standard is necessary, including data standard, analysis algorithm standard and software design standard. Standard design is a key content of remote collaborative fault diagnosis. Flourishing Web technology makes remote collaboration practical, and there are some practical fault diagnosis systems that are just based on remote collaborative technology now [8]. For example, grid technology is just an advanced remote collaborative technology.

## 3. Grid-Enabled Remote Fault Diagnosis System

### 3.1. Grid Technology and Fault Diagnosis

The grid is a name that was first coined in the mid-1990s to denote a proposed distributed computing infrastructure for advanced science and engineering projects. Grid technology is a growing information technology, where that the main purpose is to realize resource sharing and collaborative working in virtual network environment, which can eliminate the information island and resource island. As explained by [1], grid should enable coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organization.

Different with the traditional Internet technology which achieves the hardware connection, grid technology organizes and manages all of the computational resources, data resources and storage resources, to set up a completely open environment for

computation and providing services, and finishes the jobs by distributing the jobs, cooperating the resources and returning the result automatically.

The new character of grid is combining Service-Oriented Architecture (SOA) concepts and Web Services technologies to create Open Grid Services Architecture (OGSA)<sup>1</sup>. OGSA is an important propositional standard by Global Grid Forum (GGF), and it is important grid architecture after the early layered grid architecture. The object of OGSA is to realize resource management and service sharing cross heterogeneous platform. OGSA defines Grid Service, which is a special kind of Web Service, providing service discovery, dynamically service creation, lifecycle management, notification, and so on [9][10].

The Open Grid Services Infrastructure (OGSI) service specification is the keystone in implementing this architecture, followed by the recent Web Service Resource Framework (WSRF)<sup>2</sup>, which is a set of six Web Services specifications that try to meld Web Service with grid computing by defining how to model and manage state in a Web Service context. Defined by OGSA, a Grid Service is basically a Web Service, which is a set of Internet-based distributed processes. Based on standards such as XML, SOAP, WSDL and UDDI, the promise of Grid Services is to enable a distributed environment in which any number of applications, or application components, can interoperate seamlessly among organizations on the grid.

Fault diagnosis technology has developed from centralized to distributed, and from Internet-based to collaborative. In manufacturing field, especially as the concept of advance manufacturing and digital manufacturing are put forward, manufacturing equipment has been developed to large-scale and highly complicated. Fault diagnosis technology for modern manufacturing equipment has faced with some challenges, and no one single system can solve fault diagnosis problem [11].

As we demonstrated in Subsection 2.3, since that grid is one kind of advanced collaborative technology, some people have noticed the potential to solve fault diagnosis problem using grid technology [3][8]. Grid technology exactly meets the requirements of fault diagnosis. Different manufacturers and experts could be integrated and connected to build a common diagnosis service sharing platform, to eliminate the information island. It is undoubtedly that the research and development of grid technology has provided a better foundation for fault diagnosis in manufacturing engineering field.

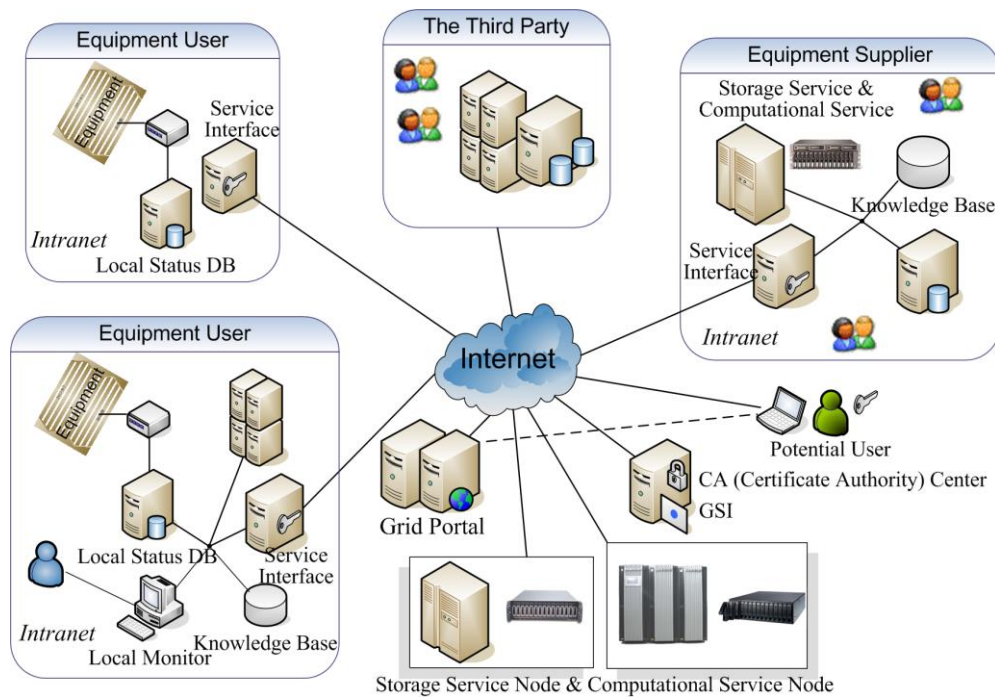
### 3.2. Architecture of Fault Diagnosis Grid

Since the development of grid technology makes it possible to solve the problem, and meets the requirement of remote fault diagnosis, in this section a framework of remote fault diagnosis for manufacturing equipments based on grid is proposed, which integrates related technical resources in equipment suppliers, equipment users, equipment manufacturers, component suppliers, equipment partners, *etc.*, where services are deployed to grid nodes. These grid resource nodes are administrated and scheduled by grid platform, and they coordinate each other and share resource information, in order to provide service for grid users. Figure 1 shows the architecture of fault diagnosis system based on grid, and it is not a hierarchical architecture, but a networked star architecture. There are some important concepts and parts in this framework, and they are listed as follows.

---

<sup>1</sup> <http://www.globus.org/ogsa/>

<sup>2</sup> <http://www.globus.org/wsrf/>



**Figure 1. Architecture of Fault Diagnosis System based on Grid**

- (1) *Intranet and Internet.* Each enterprise has its Intranet, and the Intranet is connected to the Internet.
- (2) *Manufacturing equipment.* Equipment user buys manufacturing equipment form equipment supplier. The equipment is monitored through all kinds of sensors, such as hydraulic, optical, electrical, fiber grating sensors, etc.
- (3) *Equipment supplier.* Equipment supplier provides one kind of or a series of equipments. The supplier also provides storage service and computational service, and these services are encapsulated using standard interface, which specify stateful Web Services and are described using Web Service Description Language (WSDL). The equipment supplier maintains a knowledge base. A *Worker Node* is deployed here in charge of fault diagnosis computing service.
- (4) *Equipment user.* Equipment user buys the equipments and obtains technology support from equipment supplier. Equipment user executes local monitor, manages a local status database, and maintains a knowledge base, too. It also provides storage service in local. A *Client Node* is deployed here.
- (5) *Storage service node.* This node is provided by enterprise partner or institute partner, which just provides a large-scale storage space for equipment status data or fault data.
- (6) *Computational service node.* This node is just providing complicated computation function and computational analysis software or algorithm by enterprise partner or institute partner.
- (7) *The third party.* It means any enterprise partner or institute partner. It can provide storage service, computational service or fault diagnosis service. The equipment status data are available for the third party. A *Worker Node* can be deployed here.
- (8) *Certificate authority center.* It is also known as CA center. All reliable servers and users are given a secure standard X.509 digital certificate, which makes sure a Grid Security Infrastructure (GSI).
- (9) *Grid portal.* Grid portal provides a user-friendly access to resource in grid environment. Portal should be responsible for job distribution, job management, resource scheduling, and data management. Users are allowed to single signed-on

(SSO) to access heterogeneous, autonomous and multiple administrative domain resources. A *Master Node* is deployed here.

From the grid point of view, everything is service, and grid enables service integration. In fault diagnosis system based on grid, there are two main kinds of service integrations, one is storage service integration, and the other is fault diagnosis service integration. To support the complicated fault diagnosis, it requires integrating modeling, estimation and analysis tools for fault diagnosis on the grid. The service-oriented architecture is adopted to support this integration and it is defined to meet the specifications of OGSA and WSRF. In this system, we integrate all the resources and information in every grid node, where the needed services are expressed as 'Grid Services'.

As you can see in Figure 1, all of these organizations can get benefits from this system, which can reduce the cost and enhance the efficiency of fault diagnosis for equipment supplier, and elevate the equipment reliability for equipment user, and provide fault data to the third party for research. In addition, equipment supplier will obtain economic profit through maintaining this system. The usage steps of fault diagnosis grid are as follows: 1) Resource and service registration. Manufacturing equipment suppliers, equipment users, and domain experts register their resources and services to the grid platform through Internet. After authorized by the grid, they become grid nodes. 2) Fault diagnosis service request. 3) Fault diagnosis service invoking.

### 3.3. Features of the Proposed System

The proposed fault diagnosis system based on grid technology has the following characteristics:

- *Commonality*. The low level data communication and processing platform has the commonality. Thus, it can be applied to a wide range of machinery equipments, manufacturing equipments, industrial machinery, such as rotating machinery, vibration machinery, and vehicles.
- *Transparency*. From the grid point of view, the details of back-end techniques are hidden and transparent to the users who do not care about where the resources are and how they are integrated in the back-end.
- *Service mode*. The usage pattern of the fault diagnosis services in grid environment is that the diagnostic task requests are submitted by users through the Web portal, and then users wait for diagnostic results.
- *Scalability*. New diagnostic resources can be dynamically added, such as new computing nodes, new diagnosis expert databases, or case-databases, reasoning knowledge and other resources.
- *Standardization*. All the grid services are developed to follow well-known grid and web standards, so that services in the diagnosis system can be flexibly consumed by clients.

## 4. Implementation of Prototype System

We implemented a fault diagnosis prototype system, in order to validate data-driven grid-based manufacturing equipment fault diagnosis method proposed in this paper, through the share of computing resources, storage resources and diagnostic programs. Fault data is collected, saved, and then submitted to the fault diagnosis grid for further analysis. In this section, we first introduce an important data management middleware BitDew used in the prototype system, and then describe the overall system design and Client-Master-Worker computation model. After that, we introduce the implementation of major modules including Web portal, user management, job management, data management and resource registration. Finally, grid fault diagnosis system is deployed on

multiple nodes in network environment to achieve collaborative computing, and the speedup performance evaluation results are given.

#### 4.1. Data-Driven Data Management Middleware

Generally, fault data are unstructured data, and therefore we can not use common relational database management system (DBMS), such as MySQL, SQL Server, *etc.* Fault data are suitable for store in separate files. File size is not a fixed value, and file size depends on the measurement time. The amount of data may be very large, that varies from a few Megabytes to several Gigabytes. Data store and transmission should follow these requirements: 1) *the reliability of data storage*, that a certain number of storage node failures are allowed; 2) *data availability*, that trusted users can arbitrarily access data in scope of a trusted domain; 3) *the reliability of data transmission*, that data is safely transmitted from the source to the destination. In a word, data is the core of fault diagnosis. Ref. [12] also presented data driven and knowledge-based techniques in fault diagnosis and fault tolerant control. Therefore, we adopt a data-driven data management middleware to overcome these challenges.

BitDew<sup>3</sup> is an open source data management middleware under the GPL which provides a programmable environment for cloud, grid and desktop grid, which has been developed by INRIA [13]. BitDew proposes several attribute keys to control data distribution, and offers programmers a simple API for creating, accessing, storing and moving data with ease, even on highly dynamic and volatile environments. BitDew middleware is written in Java using the RMI technology, which can run on a variety of operating system platforms. The software architecture of BitDew is composed of two layers: distributed application development layer, BitDew run-time layer (API, and core services). It provides a set of simple API library to facilitate user-friendly program development. BitDew is a data-driven middleware that fully supports data-intensive applications. It has been adopted in some applications, such as large-scale information processing, bioinformatics [14], optical character recognition [15], and content distribution network.

BitDew relies on the following six low-level *Attribute* values of *Data* object to realize a user controllable and transparent data scheduling and distribution algorithm:

- *replica*, which indicates how many occurrences of a data should be available at the same time;
- *fault-tolerance*, is a flag which controls the policy in presence of machine crash;
- *lifetime*, is an attribute absolute or relative to the existence of other data, which decides the lifecycle of a data in the system;
- *affinity*, which drives movement of data according to dependency rules;
- *transfer-protocol*, which gives the runtime environment hints about the protocol to distribute the data;
- *distrib*, which indicates the maximum number of copies of *Data* with the same *Attribute* should be sent to particular node.

A typical example of metadata and the attributes definition are given as following:

```
Attribute attr = { replica=2, fault-tolerance="true",  
                 lifetime="2013-11-01 00:00:00",  
                 affinity=dataA.getUid(), protocol="http", distrib=1 };  
Data dataB= new Data(file, attr);
```

Due to the open and programmable environment, it is easy to implement a new data transfer protocol in BitDew for developers. BitDew now supports a variety of transfer protocols, such as *http*, *ftp*, *bittorrent*, and more recent *Amazon S3*. Data management and

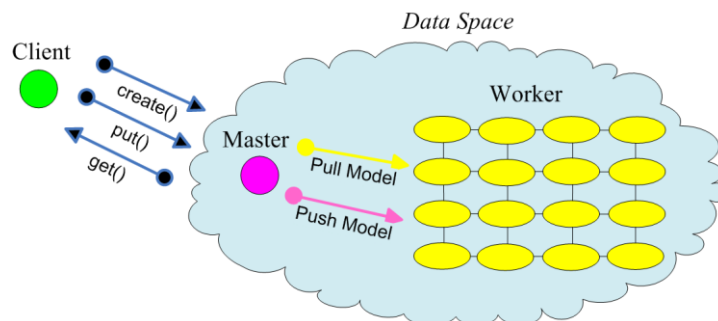
---

<sup>3</sup> <http://www.bitdew.net>



distribution in BitDew relies on four core services: 1) *Data Catalog* (DC), provides data directory service to save metadata information, such as name, size, MD5 checksum; 2) *Data Repositories* (DR), is a file storage services, and manages large-scale data upload and download requests; 3) *Data Transfer* (DT), provides data transmission services, and starts automatically the external transmission tools, such as *http* and *ftp* transfer tool, and monitors the process of the transfer; 4) *Data Scheduler* (DS), provides dynamic data scheduling service, according to the definition of metadata attributes, and controls data should be stored in which nodes.

BitDew provides a common Master/Worker mode of communication that one master node interact with multiple worker nodes using BitDew. BitDew-based application development must be in accordance with the following steps. First, we start BitDew core service DC, DR, DT and DS. Second, we create a data object, and define the attributes of the data. Third, the data object is stored in the data space, waiting to be scheduled. Fourth, we write programs that run at workers when workers receive a data. Fifth, master node collects the result from workers. Sometimes, we can also add the Client to Master/Worker mode, as you can see the Client-Master-Worker mode in Figure 2.



**Figure 2. Client-Master-Worker Communications and BitDew Data Space Model**

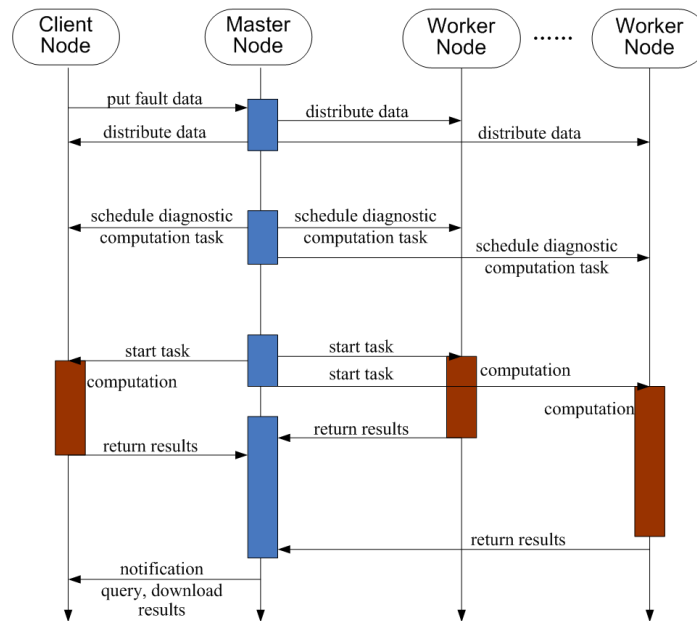
In order to deal effectively with data storage and transmission, the data space model is proposed to manage data. The data space has the following characteristics: 1) it provides a virtual storage environment, as well as metadata management; 2) it provides methods to data operations, including *create* method, *delete* method, *put* method, and *get* method. Here, we detail the Client-Master-Worker communications in BitDew. The communications among Client, Master, and Worker are shown in Figure 2. Client communicates with data space through *create*, *put* and *get* methods; Master runs at reliable node which is responsible for data management and scheduling; data synchronization is realized by heartbeat mechanism between Mater and Worker; Master uses the “*push*” mode to push data to Workers; Workers use the “*pull*” mode to obtain data from the Master.

#### 4.2. Client-Master-Worker Computation Model and Fault Diagnosis Workflow

As we mentioned before, equipment user can be deployed as a Client Node, and the third party and equipment supplier can be deployed as a Worker Node, and there is also a unique Master Node which is usually deployed in accompany with the grid portal. The whole system works in accordance with the Client-Master-Worker architecture [14]. Client Node submits task; Master Node schedules tasks, and manages Worker Nodes to provide services.

- (1) *Client Node*. The node connected to the diagnosis system that hopes to consume a set of diagnosis services is the Client Node. Client Nodes are the main users in the system.

- (2) *Master Node*. It is the system management node. The core services are deployed in the Master Node, which is responsible for managing the resources, scheduling and controlling the execution of computation tasks, managing the behavior of all Worker Nodes, and building a Web server to accommodate all operations of users through the grid portal on the Master Node.
- (3) *Worker Node*. It is the node dynamically added to the fault diagnosis grid, which provides storage and computational services and interface, under the unified management.



**Figure 3. Dataflow Scheduling in Fault Diagnosis System based on Data-Driven Grid Computing**

Since BitDew is used as data-driven data management middleware, the core of fault diagnosis grid is the dataflow scheduling algorithm, which is shown in Figure 3. The complete process in this system, from the acquisition of equipment fault data until the results of fault diagnosis are obtained, is divided into the following six main steps:

- **Step 1:** When equipment goes wrong, the monitoring device uploads the equipment status data to Master Node from Client Node. Thus, we integrate the storage resources in this system to build a distributed file system which is managed by Master Node, and then it is easier to realize the storage of large-scale fault and status data. A large data may be divided into a number of small data pieces that are respectively stored in different storage nodes.
- **Step 2:** Worker Nodes provide the fault diagnosis services, which encapsulate diagnostic programs. After the fault data have been uploaded, according to the diagnostic rules, Master Node chooses the right program to analyze status data.
- **Step 3:** Master Node starts to distribute the task and data to other nodes where the computational services are deployed, and the computational resources in those nodes are used to compute, through starting the chosen diagnostic program.
- **Step 4:** Those Worker Nodes return the results to Master Node one by one, and finally, Master Node sums up the computational results of those nodes and achieves the final diagnostic result.
- **Step 5:** The final result is stored in the distributed file system.
- **Step 6:** Legal authorized users from any nodes can access the diagnostic result which can be displayed for inquiry and downloading.

### 4.3. Module Implementation

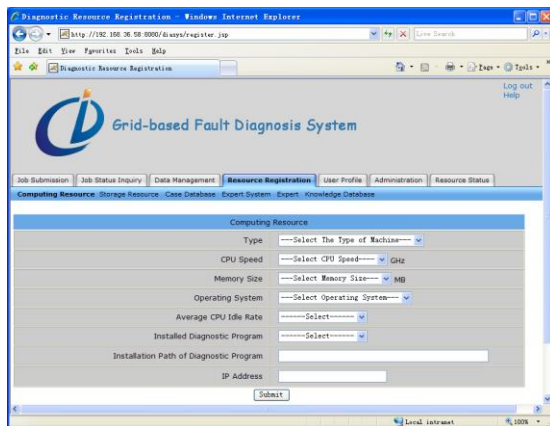
The prototype system has been developed using BitDew middleware. It consists of some components, including service container, Web portal, Worker program, Master program. It is a GT4 WSRF-based standard, scalable and extensible platform. The user-friendly Web portal is developed using Java Server Page technology. System development tools include JDK1.6.0\_06, Eclipse-SDK-3.3, and bitdew-sdk-0.2.2. System deployment environments are as follows: MySQL-5.0.19 is used as the DBMS; Apache Tomcat-6.0.16 is launched as web container. In accordance with BitDew pattern, the fault diagnosis grid is built mainly using a set of nodes, among which, one node as the portal to run Web server; one node as Master to run data management service; other nodes as Workers to run computing services.

Majority of data are time-domain signals. Acquisition time interval and acquisition accuracy have a direct impact on the amount of data. Generally, it requires analyzing and processing time-domain signals, using a variety of different transforms. The commonly used methods include amplitude-domain analysis, time-domain analysis, frequency-domain analysis, power spectrum analysis, and so on. As an important signal analysis tool, Fast Fourier Transform (FFT) is widely used to extract the spectral characteristics of the signal. In the prototype system, we use FFT to simulate the diagnostic calculation program.

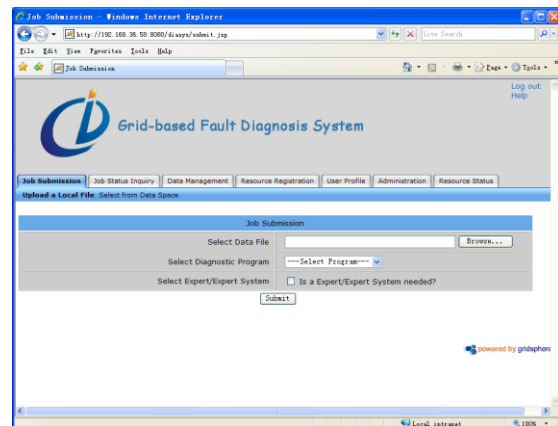
We generated state data and fault data by emulation the running of devices. Fault diagnostic program first unpacks and parses fault data, and then performs the FFT computation. Results are returned when all calculations are completed. The users first log into the prototype system, then register resources, submit data, submit task, query task status, finally download the diagnostic results. Three typical function modules and their implementations are demonstrated as follows:

#### (1) Resource Manager

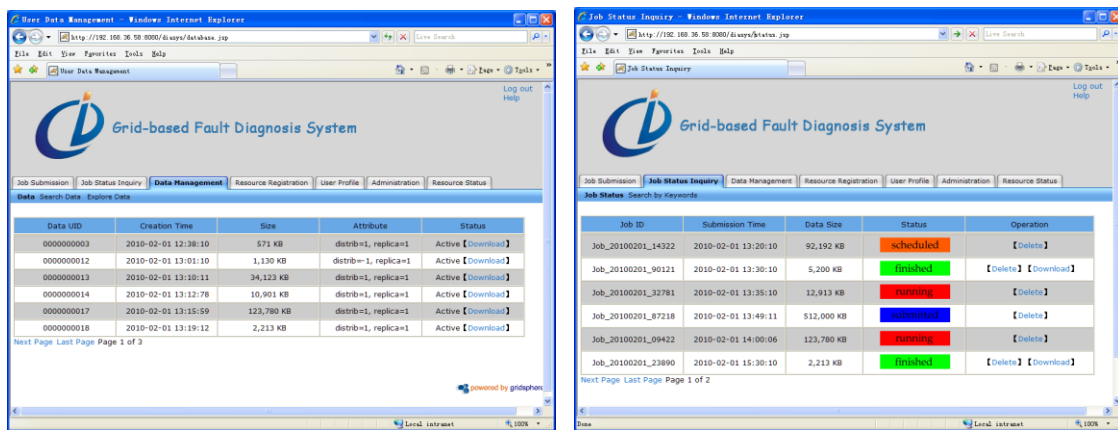
Resource Manager is responsible for resource registration and diagnosis computing resources encapsulation. The users login through the grid portal of the system, and then register computing resources. The diagnosis computing resources registration web page is shown in Figure 4(a). The details of computing resources are inserted, such as compute node type, CPU speed, memory size, the average CPU idle rate. It must also specify the path of installed diagnostic calculation program and the working path. Once the registration is successful, the new resource will be started as a Worker, so that computing tasks will be assigned to it.



(a) Diagnosis Computing Resources Registration



(b) Diagnostic Job Submission



(c) Diagnostic Job Status Inquiry

(d) Data Management

**Figure 4. The Web Portal of Fault Diagnosis System**

## (2) Job Manager

Job Manager is responsible for job management and job status monitor. Users can select a local file, then upload it to the data space, simultaneously submit tasks at the same time, while they can also directly select the data that already exists in the data space and then submit directly. The diagnostic job submission web page is shown in Figure 4(b). The diagnostic job is represented by job ID, submission time, data size, and the current status. The status of the job is represented by the state of the data. After the submission of a job, the fault data is first-come-first-served (FCFS), and migrated within four states: *submitted*, *scheduled*, *running* and *finished*. When the job status becomes *completed*, the user can download the results. Figure 4(c) shows the job status querying web page. In this prototype system, BitDew data management middleware is adapted to separate big data into a number of small blocks. The status of the job that has been *completed* shall be turned into green, which means that the corresponding analysis and calculations of all data blocks have been completed.

## (3) Data Manager

After the user submits diagnostic tasks, the corresponding data can be retrieved in the data space. The diagnostic results are also saved to the data space when the task is finished, which can also be saved to local through the download link. As is shown in the data management web page in Figure 4(d), metadata information are listed in this figure, such as data UID, creation time, data size, attributes, status, *etc.* The two attributes of the data, *distrib* and *replica*, determine the number of copies.

## 4.4. Performance Evaluation

In order to evaluate the performance, the prototype system is deployed using 13 machines in a campus network environment. Table 1 specifies machine hardware configuration for these 13 machines. Site B and Site C are in the same laboratory, while Site A locates in another campus. The actual average network bandwidth between two sites is shown in Table 2.

**Table 1. Hardware Configuration of Machines**

Site	Node Number	Type	CPU	Memory	OS	Network
A	8	Cluster	AMD Operon 2.2GHz	2GB	Linux RedHat AS3	1Gbps
B	4	PC	Intel P4 2.93GHz	512MB	Windows XP	100Mbps
C	1	PC	Intel Dual Core 1.66GHz	1GB	Windows 7	100Mbps

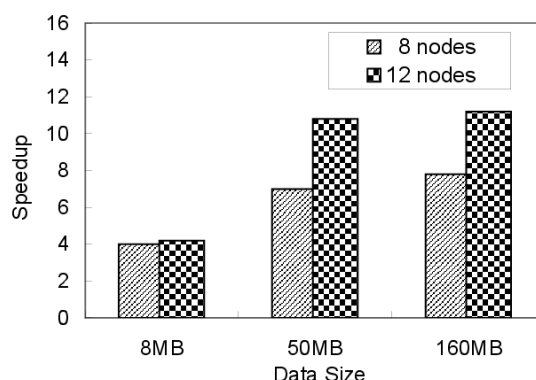
**Table 2. The Average Network Bandwidth between Two Sites**

Two Sites	The average data transfer rate
A ↔ A	31.5MBps
A ↔ B/C	6.1MBps
B ↔ B/C	9.3MBps

One node of Site A runs as the server to deploy Web server, and starts BitDew code services (DC, DT, DR, DS) and Master program, and the remaining 12 nodes run as Worker to start computation program. Users login through the Web portal and then submit diagnostic tasks and upload data. In our evaluation, we measured the time of task accomplish time, including data downloading time and execution time. We ignored the system overhead, such as Master scheduling time delay, Worker response time delay, downloading delay and other tiny overhead. We used data unpacking and Fast Fourier Transform (FFT) calculations to simulate the diagnostic analysis application. We generated some data samples for our evaluation. The size and samples number of fault data are shown in the following Table 3.

**Table 3. The Details of Fault Data used for Evaluation, Including the Data Size and Samples Number**

#group	Data Size	Samples Number
1	8MB	1, 048, 576
2	50MB	5, 242, 880
3	160MB	20, 971, 520



**Figure 5. Speedup in Data-Driven Grid Computing with Multi-Worker Nodes**

We made a comparison of the total time consumed to complete the diagnostic task, in the case of single node and multi-nodes approach. The single node approach means without grid technology, while the multi-nodes approach is using grid technology. For the three groups of diagnostic fault data, when the number of computing nodes is 1, 8 and 12,

respectively, we measured the completion time of task. The speedup is used as a metric for our evaluation. The speedup is the ratio of the time of single node approach to the time of multi-nodes approach. Data block size is set to 2 MB in our evaluation. For the three groups of data, whose size are 8MB, 50MB and 160MB, the task completion time using single node approach is 9.2s, 78.1s and 254s, respectively. The speedup of using multi-nodes approach is shown in Figure 5. It can be seen from the results that the speedup is close to the theoretical linear value. In the case of using 12 computing nodes, task completion time is decreased by 76.1%, 90.8% and 91.1%, respectively, compared with using single node. Thus, adopting data-driven grid technology, through collaborative computing of a plenty of resources, it can significantly speeds up the calculation and analysis of the fault data.

## 5. Conclusion

On the basis of grid technology and Internet-based remote fault diagnosis technology, this paper proposed an integrated fault diagnosis system based on grid to meet the demand of remote fault diagnosis for modern complicate equipments. The architecture of fault diagnosis grid has been given, which provides service integration methods to integrate equipment suppliers, equipment users, the third parties, and all kinds of storage and computational services. This open system advances the efficiency and capability of fault diagnosis, and also allows sharing the diagnosis services that makes this system more extensible. Besides, we discussed the diagnosis workflow in data-driven grid computing environment. Using BitDew data management middleware, we implemented a data-driven fault diagnosis grid prototype system, which has been deployed in three different sites to evaluate the performance. It is undoubtedly that fault diagnosis based on grid will be a revolutionary technology in the future.

## Acknowledgements

This work is partly supported by the Natural Science Foundation of Hunan Province under grant no. 2015JJ3071, and Scientific Research Fund of Hunan Provincial Education Department under grant no. 12C0121.

## References

- [1] I. Foster and C. Kesselman, "The Grid 2: Blueprint for a new computing infrastructure", USA: Morgan Kaufmann (2003).
- [2] X. Ren, M. Ong, G. Allan, V. Kadiramanathan, H.A. Thompson and P.J. Fleming, "Integrated fault diagnostics on the grid", Proceedings of the 9th International Conference on Engineering of Complex Computer Systems, (2004), pp. 59-65.
- [3] X. Ren, M. Ong, G. Allan, V. Kadiramanathan, H.A. Thompson and P.J. Fleming, "Service-oriented architecture on the grid for integrated fault diagnostics", Concurrency and Computation: Practice and Experience, vol. 19, no. 2, (2007), pp. 223-234.
- [4] M. Ong, M. Alkarouri, X. Ren, G. Allan, V. Kadiramanathan, H.A. Thompson and P.J. Fleming, "Grid-based decision support with pro-active mobile computing", Proceedings of the 2005 IEEE International Conference on Services Computing, (2005), pp. 59-66.
- [5] L. Zhang, C. Yuan, Y. Cheng and Z. Zhou, "On key technologies for the development of remote diagnosis system of an equipment", Journal of Huazhong University of Science and Technology, vol. 27, no. 12, (1999), pp. 50-52.
- [6] X. Yuan, X. Li and L. Qu, "Design of the Internet-based machinery remote diagnosis center", China Mechanical Engineering, vol. 13, no. 10, (2002), pp. 874-876.
- [7] W. Wu, S. Yang and J. Wu, "The design and implementation of remote cooperative diagnosis system", Computer Engineering and Applications, vol. 36, no. 1, (2000), pp. 139-141.
- [8] S. Cao, Y. Chen, Y. Han and M. Fang, "Research on remote service and fault diagnosis system based on grid technology", Computer Integrated Manufacturing System, vol. 10, no. 12, (2004), pp. 1577-1582.
- [9] I. Foster and C. Kesselman, "Globus: a metacomputing infrastructure toolkit", International Journal of Supercomputer Application and High Performance Computing, vol. 11, no. 2, (1997), pp. 115-128.

- [10] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the grid: enabling scalable virtual organizations", *International Journal of High Performance Computing Applications*, vol. 15, no. 3, (2001), pp. 200-222.
- [11] R. Zhang, Y. Sun and J. Zhang, "Research on remote distributed intelligent monitoring, diagnosis, maintenance system for special engineering vehicle", *China Mechanical Engineering*, vol. 13, no. 12, (2002), pp. 1028-1031.
- [12] H. Wang, T. Chai, J. Ding and M. BROWN, "Data driven fault diagnosis and fault tolerant control: some advances and possible new directions", *Acta Automatica Sinica*, vol. 35, no. 5, (2009), pp. 739-747.
- [13] G. Fedak, H. He and F. Cappello, "BitDew: A data management and distribution service with multi-protocol file transfer and metadata abstraction", *Journal of Network and Computer Applications*, Vol. 32, No. 5, (2009), pp. 961-975.
- [14] H. He, G. Fedak, B. Tang and F. Cappello, "BLAST Application with data-aware desktop grid middleware", *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, (2009), pp. 284-291.
- [15] M. Labidi, B. Tang, G. Fedak and M. Jemni, "Scheduling Data and Task on Data-Driven Master/Worker Platform", *Proceedings of the 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, (2012), pp. 593-598.

## Authors



**Bing Tang**, Ph.D. and Lecturer. His research interests include parallel and distributed system, cloud computing. He is a member of CCF, IEEE and ACM.



**Li Zhang**, Master and Lecturer. Her research interests include parallel and distributed system, cloud computing.

