[redacted]

Open Knowledge and SmartFog
The Agent-based Datacenter Self-management

Abstract

At present, with the development of the network based systems; the datacenter is increasingly tremendous and complicated. The data management confronts a number of problems generated by the instability and scaling. According to the status, some solutions for simplifying large-scale management are required. This proposal presents the design for an open knowledge based modular framework for data center management. The design processes the data provided by individual virtual machine from the SmartFog based control algorithms, which make the self-management possible.

It is not clear from the abstract what will be involved in this work. The student should explain (briefly!) what is novel about the new 'design' they mention.

Motivation

Nowadays, the datacenter provides support for running multiple data model in different ways to share the underlying resources. Originally, data management designed as centralized model for isolating simple applications. However, the scale and complexity of datacenter causes a number of problems for management. Therefore, convenient management models are demanded.

Developers have interacted the distributed model with data management (as the could computing). As a component based open system (OpenKnowledge system) and a component based frame (SmartFog), they have the feature of distribution. With convenient component use and flexible structure, more distributed management solution is possible to be built on the SmartFog&openKnowledge system.

When introducing the research problem, try to provide very clear examples of the issue, why it is difficult and the implications of a potential solution. Here the student only says "...causes a number or problems..." but does not explain what they are. A clear explanation with a few solid real-world examples will earn you credit in this section.

Background

Open Knowledge

There are plenty of different initiatives of open knowledge existing, such as the open knowledge network, which is active in order to communicate and collect local knowledge. According to the open knowledge network application, it can be seen that the network is supported by a number of information and techniques. The open knowledge network is structured on some basic concepts, which can be described as follow: "1. Building capacity in communities to support knowledge sharing. 2. Working offline for free, but synchronizing with the Net. 3. Peer-to-peer networking of existing Knowledge Workers. 4. Standards for metadata using XML. 5. Building on the experience of others. 6. Agreed open content copyright licenses. 7. Sustainable business models adapted to different contexts. "[15]

The open knowledge approach is more suitable than the web-services approach in

Lists should not be mixed in with the paragraphs; should be formatted better. The student just copied this list as a quote from a source then moved on. Instead, they should take time to explain the meaning of each point in the list. It is not enough to just make a list here.

some aspects. However, it is much stricter in some other aspects. Semantic Web services aims at automatic framing simple solution to complex on-line services with intelligent algorithm, while the open knowledge of services pays more attention to the implementation of predefined "workflow". The Open Knowledge only decides in the run-time instances for the implementation of a service that means the agent services provided will be used.

The Open Knowledge recruitment is more common than the Web services architecture, because the service and the recruit execute separately. In the Web services architecture, it is generally believed that the matching goals of both approaches are the same, while a combination of objectives is in two different ways. Open knowledge aims to operate predefined work-flows, while semantic web-services aim to interact complicated work-flow for avoiding trivial work. Open knowledge suits to be a distributed storage model completely. However, most dominant web-service frameworks set a centralized architecture. [18]

"Matching is a critical element in many application domains. The ontology matching has been understood in traditional applications. We view ontology matching as the process that takes as input at least two formal structures, generically called ontologies [16]." In the area of open knowledge project, developers have framed two ontology matching technologies. They make tools operate on the open platforms. One algorithm, PowerMap is the core mapping component of the PowerAqua question answering system [2]. It selects the results of user queries dynamically. PowerMap is the key point of connecting user's input and the corresponding ontologies. The other algorithm is the Scarlet, which compares the concepts of ontologies with their semantic [3]. It contributes to relations by linking dynamical and extracted multiple facts from ontologies. Thus, this algorithm affords deeper level comparison.[4,5]

In general, long quotes such as this should be strictly avoided. Try to rewrite in your own words (but still show the

Nowadays, dynamicity characteristics are applied in emerging applications through peer-to-peer (P2P) information systems. Peer-to-Peer is a completely distributed communication model. In this model, every peer has equivalent capabilities in sharing data and services through a transitive P2P manner [17].

There are some advantages of P2P model in building information systems. It supports the prospects of higher scalability, lower cost and better fault resilience and aggregate performance, but in centralized framework [17]. The Peer-to-Peer framework consists of system components, which can be presented by a local information source and a local ontology. The information source affords to save data, while the local ontology builds the relation among data sharing peers. Furthermore, each peer has a host. Every host individually exists in the system. Hosts can handle in queries to the local ontology of their peers, and collect query result from related peers in the system [17].
In the statement of replying a user query, peers propagate the query, and query result sending to others. For this reason, peers set up acquaintances in the system. Therefore, while it gets a query, a peer replies it with the data from the local source

In this section, some system diagrams could have been used to illustrate some of the concepts. The student clearly has language difficulties and some of the explanations are unclear. The use of diagrams helps to convey meaning in a clearer and more universal way.

and delivers it to specific acquaintances. [17]

SmartFog

The work of configuring, deploying and managing is done via the distributed software framework named SmartFog with the implementation of java programming. It is a powerful tool to configure systems or to make systems to run through the process of setting up, starting and shutting down automatically. SmartFog has the orchestration capabilities. Therefore other functions of SmartFog include the capabilities of starting, stopping or rebooting systems in the preprogrammed orders. Additionally, it can also contribute in detecting and recovering malfunctions. [9]

The contemporary computing infrastructures feature the systems that contain multiple software networks. The functions are distributed to series of software which will work together as a united system. Thus it is essential that different software is functioning in the right order: to be started or shut separately, to be precisely configured, and to be combined as a whole. The software in a system is to be seen as a individual as well as an indispensible part. [7]

"SmartFrog consists of: A Language for defining configurations, providing powerful system modeling capabilities and an expressive notation for describing system configurations. A secure, distributed Runtime System for deploying software components and managing running software systems. A Library of SmartFrog Components that implement the SmartFrog component model and provide a wide range of services and functionality" [9]

Again, this use of long quotes gives the impression that the student is lazy or does not understand the concept well. Avoid doing this!

Smart frog is also a very powerful tool for testing applications. Within a complicated system with different layers of individual components, there is the necessity to run and examine on each of the autonomous components. Developers have used shell scripts to cope the problems of testing multiple application components. This approach is considerable to solve and test simple applications. When it comes to a more completed one, it would be of much difficulty and cost a lot of time to run and examine every individual component. Smart Frog is provided as a useful tool for developers to simplify the process by its functions discussed above. It can help the developer to save time and avoid errors. It is also a conceptual tool that allows the developer to easily locate and adjust the existing bugs.

Try to avoid repetition. Using pronouns ('it', 'they','them', etc.) if the subject is already implied makes it smoother for the reader. Alternatively the student could have made a list of 'Benefits for Developers'.

The concept of 'notation' in smartfrog is in fact to set up a system which features several open data structures. Primarily, several parsers are set up and supported by the smartfrog and can provide a variety of textual representations of the notation of open data structures. Secondarily, if the data structures are being processed as the common form, there is the possibility that GUI tools can be developed using which developers would be able to configure the components by simply 'drag and drop'. it is worth notion that no generic GUI tools have been developed for smartfrog yet. Only several

Here there should be a capital letter at the start of a new sentence. Use a grammar correction tool if available to check for such simple mistakes. Otherwise it gives the impression that you are careless and did not check or re-read your work.

experimental versions have been developed.

This whole section is full of quotes. Always re-write in your own words.

"The 'notation' is object-oriented, supporting inheritance and extension configuration instructions." [9] These descriptions include different component definition, associations, relationships and components related system life cycle and associated workflows. Descriptions of different configuration parameters with multiple instances can help to estimate correctness before an attempt is made.

The SmartFrog new version, although supporting a variety of textual language, but provides its own professional symbols "out-of-the-box." Other languages are in the process of future development. The notation is not a programming language for stating detailed behaviors. The functional part of component is to regular the programming language. SmartFog system can control the operation of components.

Nowadays, only JAVA programming interact with SmaftFog well. JAVA code does not operate on the SmartFog directly, but involve in other language for direct implement. 'The SmartFrog distribution included in the SFHOME / bin called sfDaemon script.' [9] This script will start at the host SmartFrog daemon. If users need to application code frequency, a suitable solution is to transmit to each host in some way the code is required. 'A good way is to put the JAR files on the file server, and for each host to mount a remote file server's local directory path to point to the SFUSERHOME's.' [9] When changing code on the server, each host will see all changes. With the advantage, SmartFog can make interaction with every node through the workflow.

Self-management

With the increase of data dealing amount, more intelligent management policies are demanded. Self-management in this proposal means designing framework for datacenter management, then data management operates according to the components implementing dynamic configuration to avoid a great number of detailed configurations.

Two key technologies for solving large-scale and complexity data management are Dynamic Reconfiguration and Dynamic System Domains. DR is the enabling technology behind Dynamic System Domains.

Introduce acronyms or initialisations in brackets immediately after the first use of the full term e.g. Dynamic Reconfiguration (DR)

Operating Environment, dynamic reconfiguration enables users to dynamically reconfigure, remove or install core system components into your server while the Solaris and your applications are running. Flexible partitions that allow running multiple applications and multiple copies of the OE on a single server are the dynamic system domains. For flexible resource management, Dynamic System Domains can be adapted to meet your changing application environment. [11]

Dynamic reconfiguration allows the logical sharing of resources within a single platform to achieve committed application service levels. Physical sharing involves the introduction of brand new resources into the system platform and the entire central server line. Physical sharing may require involvement from Sun service organization to avoid non-compliance with the service agreements. [12, 13]

Methodology and Techniques

The study is first hand and secondary research interacted, using published data and experiments as a source of information.

Firstly, the system framework should be considered more closely to understand the role of SmartFog in open knowledge system data management. Paul Anderson [1] provided an excellent description of modern data automatic management.
In addition, open knowledge is the key point of the design. The design makes open knowledge system as the basement of whole data system interaction and the agent-based platform.

After that, the study of configuration tools follows. The most attention must be devoted to ways it solves the problems of current framework. A brief description of configuration tools is given [19], whose paper perhaps is a good starting point of investigate the subject. The information includes configuration details, log analysis, regular expressions and connectivity. Also the comparison with alternative solutions must be done.

Finally, a new solution should be framed based on above research. The solution mainly depends on the interaction between open knowledge and SmartFog, which operates the SmartFog's workflow to control with a method that interacting every nodes to a SmartFog component.

It is not enough to simply say that a 'new solution' is required. While we do not expect you to know entirely how you will solve the problem, you should have some ideas for approaches and, in particular, how to begin. You should consult with your supervisor if you find this difficult.

The student should have included a section with potential ideas for solutions.

```
Demo1.lcc[21]
// Demo1 interaction model
r(demo1Client， initial)
r(demo1Server， necessary， 1)

a(demo1Client， ID1)::
greeting(Message) => a(demo1Server， ID2) <- demo1ClientGetGreeting (Message)
then

response(Reply) <=a(demo1Server， ID2) <- demo1ClientOisplayReply (Reply)
then
a(demo1Client， ID1)

a(demo1Server, ID2) ::
greeting(Message) <= a (demo1Client， ID1) <- demo1ServerCreateReply (Message,
Reply)
Then
response(reply) => a(demo1Client， ID1)
Then
a(demo1Server, ID2)
```

Always include a figure number and title/description

Figure 1: Example of code....

In the sheet demo1.lcc, it is the interaction model of a agent-based datacenter system. In the demo, the interaction model is written in the LCC language, which can be describes as follow:

This is a mess. Try to imagine how the reader will see your document: is it easy to read? is it formatted well? If possible, ask a friend to read over your document and tell you if it makes sense.

This just looks careless and is full of mistakes. Better/clearer formatting is required.

r(demo1Client， initial): The sentence defines the Demo1Client role is the one begins the task

r(demo1Server， necessary， 1): Defines at least 1 peer has to play the demo1Server.

a(demo1Client， ID1):: Stating the ID1 to the demo1Client '::' means that the statement of 'demo1Client' begins after it.

greeting(Message) => a(demo1Server， ID2) <- demo1ClientGetGreeting (Message): If we want to greet the definition of 'Message' it can begin to run the interaction by the 'demo1lCientGreeting (message)'. "In LCC the '<-' symbol is used to indicate that after it a constraint is defined. When the constraint is satisfied, a message 'textttask(Message)' is sent to the 'demo1Client' role identified by 'ID2'. In LCC the'=>' symbol is used to indicate that a message is sent from the current role to another role." [18]

SmartFog components can be set up in the open knowledge system as a workflow as follow:

sfShouldTerminate	should the component terminate itself (default false)
sfShouldTerminateQuietly	should the component terminate quietly (default false)
sfShouldDetach	should the component detach itself from the graph(default false)

Chart B [8]

Where is 'Chart A'? Where is the description?
Always check that your labels are consistent.
Also if a chart/diagram is from a source, make it very clear.

The workflow carries out a number of tasks, and all of them averagely live short. The workflow tries to state the all logic of component with the task in the sfStart method. It is only comfortable for flashed components. If in the complicated situation, sfTerminate component could not be called sfStart as it can create deadlock if RMI attempts to make re-entrant calls on separate threads. We recommend that every component that does this is tested inside a workflow sequence with SmartFog Should Terminate set to true, a sequence that should complete within a bounded time. This verifies that the termination is initiated.[8]

According above background and tasks, a work package is planed:

Work processes	**Expended Time**
Backgroundlearning(SmartFog& OpenKnowledge)	08 Jan- 08 Feb
Framing project proposal& Configuring OpenKonwledge system and testing	20 Feb- 31 Mar
Configuring edit environment for SmartFog and Learning components related knowledge	1 Apr – 15 Apr
Design the workflow of SmartFog and study OpenKnowledge further	15 Apr – 30 Apr
Coding with java programming to implement SmartFog framework	May - June
Interact SmartFog components and OpenKnowledge functional component in coding Testing tasks and try to make some interaction with others	May- July
Do further research based implemented tasks to frame the final disseration	Middle May- July.

Qualifications

The main constraint is that a high complexity of modern data management that requires good background knowledge in a number of subjects. To understand what is going on the physical interface some background in virtual machines and network configuration is required.[20] Obviously, programming experience is valuable, especially concerning algorithmic. Finally, some knowledge of discrete mathematics may prove useful: for example, theory of GUI is closely related to work flow, employed by communication nodes, while virtualization basics help to understand problems that appear in data centers. Despite it is necessary to dedicate some of the project time to studying these areas, as a control systems engineer, have a good background, involving these subjects. Sf problems

Partial/incomplete sentences are completely unacceptable. The student clearly did not check their work.

In addition, there are some limits on the SmartFog itself. It is a framework that interacts with other management systems for data managing. Through SmartFog can complete some tasks individually, it is not a completed solution. Therefore, in the process of development, it may frame solution-level systems with the help of other management tools, such as the LCFG [14], which is an established configuration framework for managing large numbers of UNIX workstations.[1]

Expected Results of Outline

This list of outcomes is actually good. However, the student did not explain almost any of this throughout the document. It is not enough to list what you will do, you must also explain and justify it.

- Configure the Openknowledge system and the operation environment.
- Design the interaction module between SmartFog and OpenKnowledge.
- Design the SmartFog work-flow with components.
- Implement SmartFog based functions with JAVA programming.
- Other configuration with other tools (such as LCFG).
- Configure finished tasks to datacenter to test the operational functions.
- Optimizing functional components, to make the task interact with other members and make the management distributed..

According to the status and the develop trend of scale and complexity of data management that may interact in sophisticated ways. In the situation, a high developed autonomic way to manage datacenter is necessary. The research of Paul Anderson proved that it is available to structure a configuration framework that allows different paradigm to be interacted easily, without changes to components.[1] Under the background, different solution in the same knowledge frame, such as the self-management framed by interaction between OpenKnowledge and SmartFog, is possible to implement.

References

[1] P Anderson, P Goldsack, J Paterson.(2003) *SmartFrog meets LCFG:Autonomous Reconfiguration with Central Policy Control.* Where was this published? What were the authors' institutes?

[2] Davide G, Vanessa L , Silvio P , Enrico Motta , Mathieu A ,Marta S, George A, Paolo B, David D(2008). *D8.4 - OpenKnowledge Question Answering System.* Knowledge Media Institute, The Open University, UK

[3] V Lopez, D Guidi, E Motta, S Peroni, M d'Aquin, and L Gridinoc (2008) *D8.5 - Evaluation of Semantic Web Applications.* Knowledge Media Institute, The Open University, UK

[4] F Giunchiglia, M Schorlemmer, F McNeill, A Bundy, M Marchese1, M Yatskevich, I Zaihrayeu, B Hu, V Lopez, M Sabou, J Abian, R Siebes, S Kotoulas (2009) *Comparison of Mapping Techniques.* Dept of Information and Communication Technology, University of Trento, Italy .Artificial Intelligence Research Institute, IIIA-CSIC, Spain. School of Informatics, University of Edinburgh, UK. School of Electronics and Computer Science, University of Southampton, UK. Knowledge Media Institute, the Open University, UK. Institute for Biomedical Research of Barcelona, IIBB-CSIC, Spain. Faculty of Sciences, Vrije Universiteit Amsterdam, The Netherlands

[5], L Gridinoc, D Guidi, E Motta, M d'Aquin1, G Anadiotis, P Besana, D Dupplaw (2008) *D8.2 - OpenKnowledge Semantic Browser v2* Knowledge Media Institute, The Open University, UK

[6] ODK http://www.opendefinition.org/ If you have to reference a website (in general this is not recommended), always put the date of access.

[7] *A Brief Description of the Dynamic Web Server Demonstrator* www.SmartFog.org 2005.07.20

[8]*SmartFrog Workflow* 2011
© Copyright 1998-2008 Hewlett-Packard Development Company, LP

[9]*The SmartFrog Reference Manual.* 2011
© Copyright 1998-2011 Hewlett-Packard Development Company, LP

[10] C Low, J Guijarro (2009). *A SmartFrog Tutorial.* Hewlett Packard Laboratories.

[11] *Dynamic Reconfiguration & Dynamic System Domains* http://www.sun.com/servers/highend/dr_sunfire/#g2_1

[12] B Michaelk (1998). *A reconfigurable computing primer Multimedia System*

Design

[13] *Dynamic Reconfiguration*. By Enrique Vargas - Enterprise Engineering Sun BluePrints ™ OnLine - April 1999. http://www.sun.com/blueprints/0499/reconfig.pdf

[14] P. Anderson, A Scobie (2002) LCFG: The Next Generation. Division of Informatics University of Edinburgh

[15] *The Open Knowledge Network,* Copyright © 2011 by: Open Knowledge http://www.openknowledge.net/

[16] P. Shvaiko and J. Euzenat(2005). *Ontology matching: virtual thematic information resource.* www.OntologyMatching.org.

[17] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu(2002). *Peer-to-Peer computing.* Technical report, HP Laboratories, Palo Alto.

[18] R Siebes, D Dupplaw, S Kotoulas, A Perreau, d Pinninck, F. Harmelen, and D Robertson (2008). The OpenKnowledge System: An Interaction-Centered Approach to Knowledge Sharing

[19]P. Anderson. (2007) Sage- System Configuration. USENIX Association

[20] M Schmid, D Marinescu, R Kroeger (2008). A Framework for Autonomic Performance Management of Virtual Machine-Based Services.

[21] demo1.lcc afforded by the Supervisor Paul Anderson

Mention this as a footnote on the page it was used, not in the references.

There are not enough Primary Sources here (peer-reviewed articles, papers, journals, etc.). Referencing websites, if not absolutely essential, should be avoided as often they are not considered academic sources. If you are unsure, then you can ask me or your supervisor.

GENERAL COMMENTS

The language in this work is very bad. I have not corrected the numerous spelling, grammar and semantic errors as there are so many. As such, please do not assume that any of it is correct.

This student clearly has difficulties with technical English writing. We understand that this can be difficult for non-native speakers and we will tolerate some errors. However, this particular work appears careless and shows a lack of effort. There are many services offered by the university, such as proof-reading and various classes on technical writing (I introduced these last year, check the wiki for details). Additionally, you can ask me questions about language and make use of my opportunities to submit drafts.

Apart from this the main issue is that it is not clear what the student intends to do. This is a 'Research Proposal' and as such you MUST declare, very clearly, what you propose to do and how you intend to do it.

The work also lacks:

* Clear descriptions
* Good use of examples/diagrams
* Clean and careful formatting
* Good references and proper use of them

In general this shows a lack of care and effort. In this way, the examiner will be unlikely to forgive mistakes.

This student failed IRP.