**Knowledge Intensive Database Systems – Cloud Operations**

Meera Udani udani.m@husky.neu.edu, Kairavi Mehta mehta.kai@husky.neu.edu

**Key Words:**

Knowledge Representation, Decision Support, Cloud Operations

**Abstract:**

A Knowledge Intensive Database System is basically used for efficiently managing processes that require making critical decisions. These processes behave depending on the decision taken. This paper deals with the applications in monitoring of cloud computing services. This system helps simplifying the decision making process. Depending on the request of the client, a decision needs to be made whether to fulfil the request or deny it. Ideally, a cloud service provider has to monitor, diagnose and manage a huge number of hardware, software and platform that the cloud provides its clients with as services. We discuss in this paper how KIDS deals with such requests.

**Introduction:**

Applications these days have to manage a large amount of data and need immense knowledge to process the data. The applications may face numerous challenges in doing so. To deal with the challenges, a KIDS (Knowledge Intensive Database Systems) model, that helps the applications in an appropriate, reliable and well-structured way, was proposed. The model was based on the human behaviour to solve problems using a loop. They called it the CARE (Classification, Assessment, Resolution and Enactment) loop. This loop consists of distinct categories of knowledge that acted on distinct categories of data. The same loop, from the perspective of data, was termed as FPHD (Facts, Perceptions, Hypothesis and Directives) loop. In this paper, we discuss the application of KIDS on Cloud Operation. [1]

The most crucial requirement of a cloud operation is that it has to follow the Service Level Agreements (SLAs). The Cloud Service Provider needs to continuously monitor the services to check if it constantly meets the SLA requirements and needs to provider quicker and more optimized solutions for whenever the SLA is violated. In short, an ideal cloud operation has to monitor, diagnose and manage millions of components in the customer as well as the providers cloud. The KIDS model is used to make the above stated operations easier to work with and thus managing all cloud operations efficiently. [1]

**Background:**

The process of manually detecting the faults and violations of services are labour intensive and taxing. It requires adequate domain knowledge. It becomes unfeasible to have the problems detected and fixed manually due to the largeness and complexity of a cloud. Continuous and rapid iterations of the OODA (Observe, Orient, Decide and Act) loop [2] [3] are required for performance enhancements.

Every cloud operation needs a bid data system to handle the amount of data. The problem with that is that the data fusion through big data is inadequate. A dynamic framework is needed to transform the data from numerous sources in various formats to the format that can be read and understood by the system for efficient decision making. The framework should be able to represent data pertaining to a specific situation. [1] Features, other than attributes and basic relations, from different data sources should be included in the representations. The framework should be compatible with any kinds of reasoning engines. In short, the framework should be able to enable data fusion across various types of perceptions. [1]

A cloud operation basically focuses on the execution of the CARE loop. [1] KIDS ensure that data and knowledge captured through the CARE-loop are in a form expedient to automatic pattern recognition, machine learning, manual knowledge and tacit knowledge profiling. [2]

The KIDS solution starts with monitoring and collecting the data which is then classified into information. KIDS then applies the assessment knowledge to the acquired information to produce diagnosis. Resolution knowledge is applied on the diagnostic result to produce a directive. Enactment knowledge is then used to apply the directive which leads to changes in the system performance. The CARE loop repeats itself over and over again as and when the enactment step is performed. [2] This is a continuous process.

**Requirements:**

The requirements of the system are denoted with the help of a use case and its description

System:

Monitor Cloud Services [MCS]

The services that the end user requests from the cloud need to be monitored to make sure it meets the SLA at every point. The Cloud service provider makes use of KIDS to make sure the requirements are met.

Actors

1. Cloud Service Provider [CSP]: The CSP is the end user that provides with the cloud services to those who request for it. It is the CSPs responsibility to make sure all the requirements in the SLA are met.
2. Knowledge Intensive Database Systems [KIDS]: KIDS is used by the CSP to make sure SLAs are met. KIDS uses the OODA loop to implement the functionalities and the algorithms it uses to give a decisive output.
3. Administrator [ADMIN]: An administrator is the human-point-of-contact at the CSP that handles the customer requests.
4. Customer [CUST]: A customer is the user of the cloud services. It can be an individual or an enterprise. A customer requests for services from the CSP.

Use Cases

1. Cloud Registry [REGISTRY]: The registry handle all the new as well as the old customer requests by queuing them.

* [CUST]: Requests for services (new or updated)
* [KIDS]: Works on the incoming requests and informs CSP of the output
* [ADMIN]: Contacts the customer to either update the existing SLA or create an entirely new one

1. Cloud Management [CMGMT]: The services need to be allocated to the customer based on the requirements mentioned in the SLA. The CSP has to make sure that the services are platform independent

* [CSP]: The CSP will allocate services based on the requirements in the SLA for each customer.
* [CSP]: Provides necessary security measurements that the customer requires
* [KIDS]: resolve faults in any layer of hardware and software components that can adversely affect the SLA
* [KIDS]: Ensures information fusion from different Virtual Machines thus ensuring interoperability
* [CSP]: reallocates services based on the output got from KIDS

1. Cloud Monitoring [CMTR]: The allocated services need to be monitored to make sure that the requirements in the SLA are met at any point in time for any customer.

* [KIDS]: Monitors the services provided by the CSP keeping SLA as the reference and looks out for ambiguities
* [ADMIN]: Keeps a record of any ambiguity found in the SLA of a customer.
* [CSP]: Resolves these ambiguities

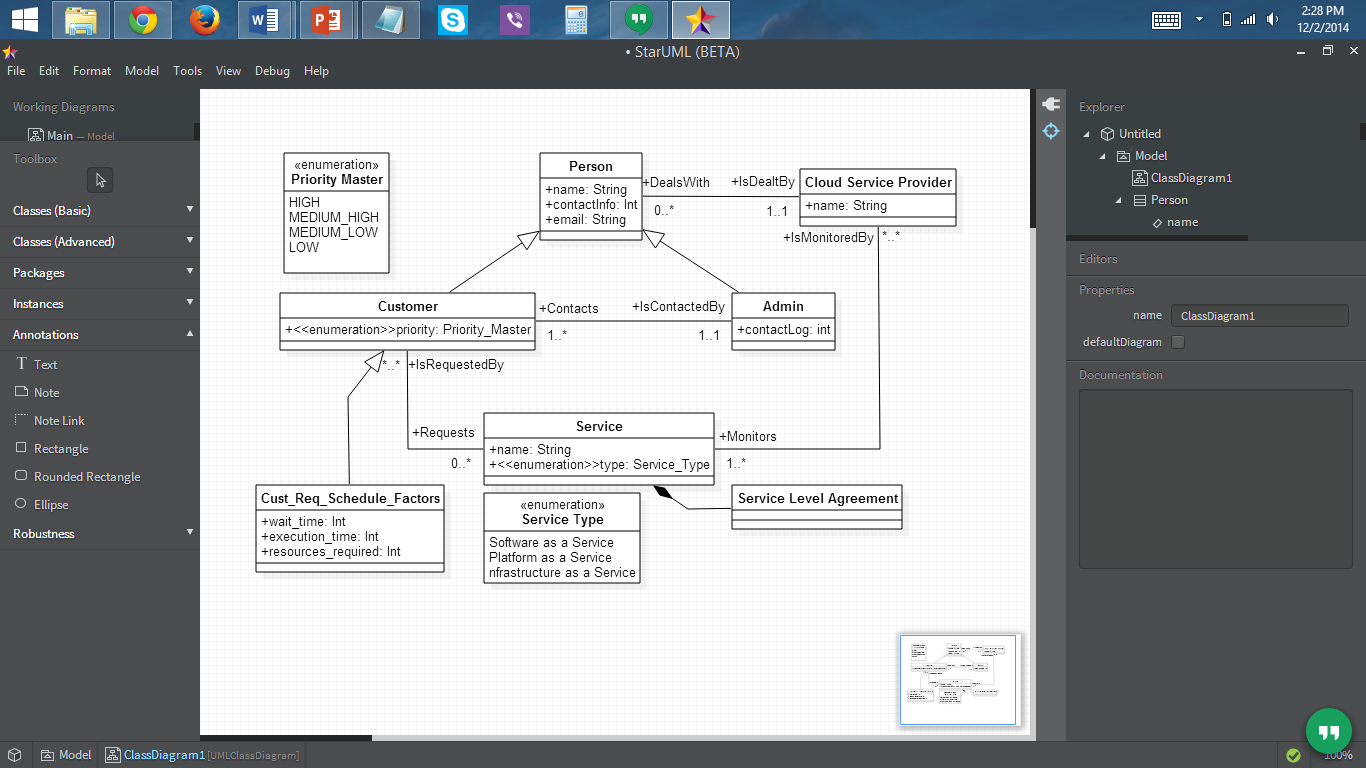
1. Reassign Services [RSRV]: The services need to be reassigned based on the customer behavior. The services can be assigned, updated or revoked.

* [CUST] : Askes for more services in which case the CSP updates the services provided
* [CUST] : may decide to switch to another CSP in which case the existing services need to be revoked
* [CSP] : Updates SLA based on the customer request

**Design:**

As per our implementation, the KIDS model handles all incoming service requests using the intelligence provided. Resource allocation is done based on the decision made by KIDS.

The class diagram for the cloud model is as follows:



The class diagram is used to develop the schema which is as follows:

* Customer Table:

CREATE TABLE "CLOUDKIDS"."CUSTOMER"

( "ID" NUMBER(\*,0) NOT NULL ENABLE,

"PRIORITY" NUMBER,

CONSTRAINT "CUSTOMER\_PK" PRIMARY KEY ("ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "USERS" ENABLE,

CONSTRAINT "CUSTOMER\_FK1" FOREIGN KEY ("ID")

REFERENCES "CLOUDKIDS"."PERSON" ("ID") ON DELETE CASCADE

ENABLE,

CONSTRAINT "CUSTOMER\_FK2" FOREIGN KEY ("PRIORITY")

REFERENCES "CLOUDKIDS"."PRIORITY\_MASTER" ("ID") ENABLE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "USERS" ;

* Cust\_Req\_Schedule\_Factors Table

CREATE TABLE "CLOUDKIDS"."CUST\_REQ\_SCHEDULE\_FACTORS"

( "ID" NUMBER NOT NULL ENABLE,

"CUST\_ID" NUMBER NOT NULL ENABLE,

"WAIT\_TIME" NUMBER NOT NULL ENABLE,

"EXECUTION\_TIME" NUMBER NOT NULL ENABLE,

"RESOURCES\_REQUIRED" NUMBER NOT NULL ENABLE,

CONSTRAINT "CUST\_REQ\_SCHEDULE\_FACTORS\_PK" PRIMARY KEY ("ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "USERS" ENABLE,

CONSTRAINT "CUST\_REQ\_SCHEDULE\_FACTORS\_FK1" FOREIGN KEY

("CUST\_ID")

REFERENCES "CLOUDKIDS"."CUSTOMER" ("ID") ON DELETE CASCADE

ENABLE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "USERS" ;

* Priority\_Master

CREATE TABLE "CLOUDKIDS"."PRIORITY\_MASTER"

( "ID" NUMBER NOT NULL ENABLE,

"PRIORITY\_LEVEL" VARCHAR2(200 BYTE) NOT NULL ENABLE,

CONSTRAINT "PRIORITY\_MASTER\_PK" PRIMARY KEY ("ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "USERS" ENABLE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

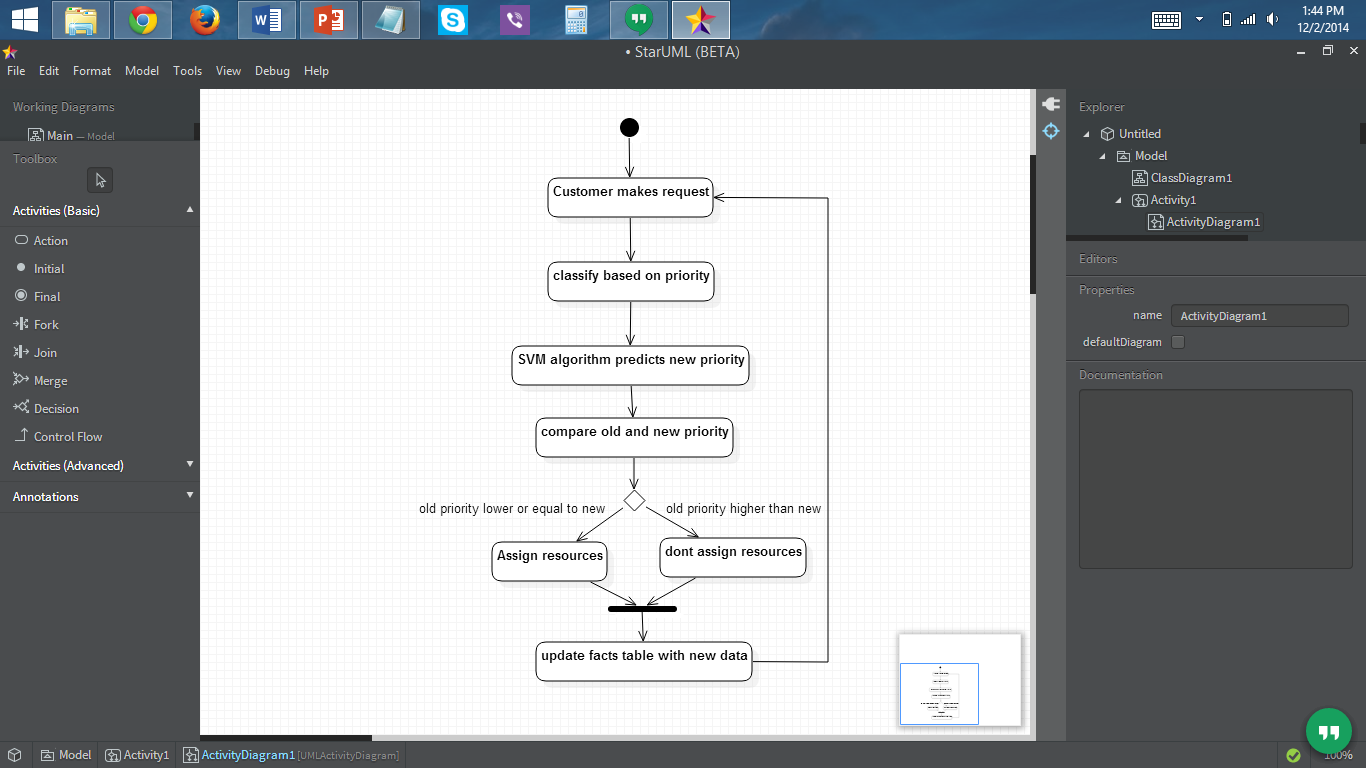
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "USERS" ;

The scenario used can be better understood with the activity diagram provided below:



**Implementation:**

* Query:
  + select cust\_id from output\_1\_6 where clas\_svm\_2\_6\_pred <= priority
  + This query is carried out on the PRIORITY\_HOPYTHESIS table whose result is stored in the NEW\_FACTS table
* Technologies used
  + Oracle Database 11g
  + Oracle Data Miner 11g R2
  + SQL Developer 4.0.3
* For diagrams
  + Star UML
* Data Mining
* Data Mining Engine – SVM

**Working:**

We have used a query to show the implementation of the system. The input is the customer request for resources. We are considering that the cloud provides with only memory allocating services and hence the customer service, for now, will consist only of memory requests. The customer is classified based on his/her priority. Priority is defined based on the past requests, wait time and other such attributes. For each request, a new priority is computed. KIDS compares both the priorities. If the old priority is less than or equal to the new priority then the resources are allocated. If it is less than the new one then resources are not allocated. The system then enacts based on the decision taken. If the resources are to be allocated, the customer is notified and the resource table is updated accordingly. After the system has enacted on the decision, it moves on to the next request and follows the same procedure. This is a continuous loop. This query follows the CARE loop procedure where the customer is classified, available resources are assessed, the collected data is resolved to make a decision and the system then enacts on the decision made.

**Conclusion:**

The KIDS model follows the OODA loop model to get the appropriate directive that is used to solve the problem. KIDS requires immense data and knowledge and means for interaction and collaboration to work efficiently. It provides with a framework to support applications that need data, knowledge and efficient management to produce adequate results. A cloud operation can consist of anything from requesting services to working on an SLA violation. With the help of KIDS, the decision making process is faster and better than when manually done. The basic framework with dummy data is developed. Future work for this project includes increasing the complexity of the project and using real time applications to test it. Also, for this project, we have assumed that the cloud provides only memory allocating services. In the future we will add more services for the customer to request.

**References:**

1. Eric S Chan, Deiter Gawlick, Adel Ghoneimy, Zhen Hua Liu, KIDS – Knowledge Intensive Database Systems, Oracle Cooperation.
2. Eric S Chan, Deiter Gawlick, Adel Ghoneimy, Zhen Hua Liu, Data, Knowledge, and Process (Knowledge Intensive Database System for Evolutionary Application Development), Oracle Cooperation.
3. Boyd, J.R., 1976. Destruction and Creation. U.S. Army Command and General Staff College.
4. Series Y: Global InformationInfrastructure, Internet Protocol Aspects and Next-generation Networks: Cloud Computing, ITU-T Y.3500 Telecommunication Standardization Sector of ITU (08/2014)
5. Huankai Chen Future Computing Group University of Kent Canterbury, UK, Frank Wang Future Computing Group University of Kent Canterbury, UK, Na Helian University of Hertfordshire Hatfield Hertfordshire, UK, A Cost-Efficient and Reliable Resource Allocation Model Based on Cellular Automaton Entropy for Cloud Project Scheduling, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 4, 2013