SDC 18

September 24-27, 2018
Santa Clara, CA

www.storagedeveloper.org

Data Driven Performance Repository to Classify and Retrieve Storage Tuning Profiles

Ismael Solis Moreno
IBM

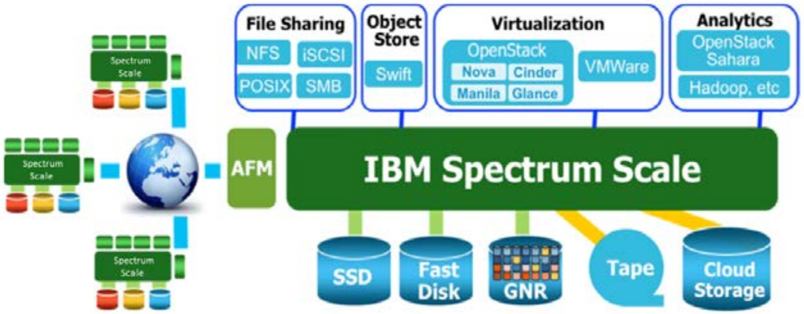
Agenda

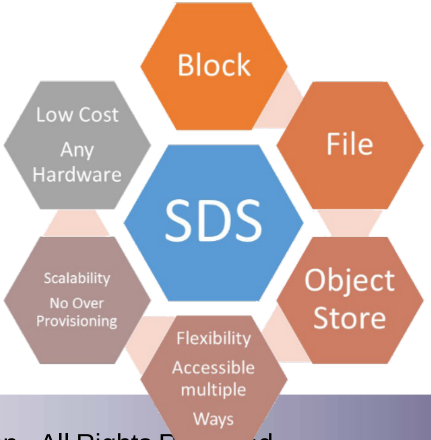
- The challenge of tuning performance for SDS.
- A data driven performance platform.
- A ML-based configuration profiles repository.
- Classifying & recovering configuration profiles.
- Comments & questions.

The challenge of tuning performance for SDS

Software-defined storage (SDS) enables users and organizations to uncouple or abstract storage resources from the underlying hardware platform for greater flexibility, efficiency and faster scalability by making storage resources programmable.

This approach enables storage resources to be an integral part of a larger software-designed data center (SDDC) architecture, in which resources can be easily automated and orchestrated rather than residing in siloes.





The challenge of tuning performance for SDS

When **administered correctly**, SDS increases performance, availability, and efficiency.

This is not so easy since SDS are normally mounted over complex environments:

- Large number of nodes
- Hardware heterogeneity
- Diverse and complex workload
- Intricate network characteristics

So, **How should I configure my SDS** solution to get the best performance, availability and efficiency?

The challenge of tuning performance for SDS

Intrinsically, SDS solutions are **very flexible**. It means that are **highly customizable**, and it means that there are **several parameters that can be adjusted** in order to provide the mentioned benefits.

So far the determination of tuning configurations are based on:

- General application best practices
- Highly skilled human experts

The challenge of tuning performance for SDS

Best practices are always a good start point but:

- Best practices normally do not cover all the possible scenarios.
- Best practices can get deprecated with the time.

On the other hand, knowledge acquired by human experts:

- Can be diminished by rotation of personnel and lack of expertise and skills sharing.

These create the need of a knowledge repository and a computing artifact to smartly classify and recover tuning profiles.

A Data Driven Performance Platform

Objectives

To research and develop a **scalable, platform to efficiently process, store, query and analyze the data from the performance evaluation of different IBM solutions**. The proposed platform must have the following characteristics:

- Provide a mechanism to ETL performance data into well defined structure to **facilitate its exploitation**.
- Results from performance benchmarks will be stored in a central repository for an **efficient access, query, and reporting**.
- **Facilitate** the latest as well as **historic view of the performance results** to the development, customer adoption, and other customer service teams teams.
- Provide an interactive user interface to **perform data mining for descriptive, predictive and cognitive analytics** (using data from different sources).
- Provide mechanisms to **create, classify and retrieve performance tuning profiles** according to the collected data on benchmark results, resource usage logs, workload and hardware characteristics.

A Data Driven Performance Approach

Project Objectives

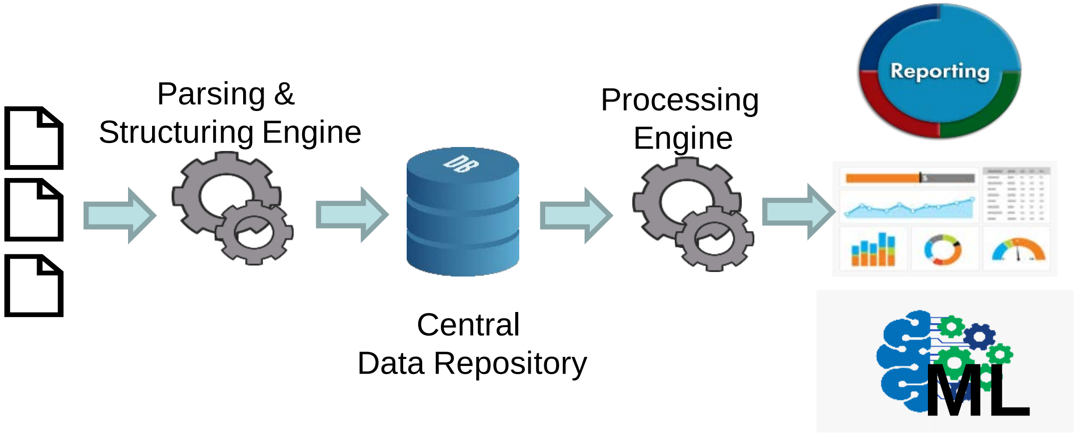
Automated Data processing

Data Storage instead of Document storage

Historic view of performance results

Performance Insights (descriptive, predictive & cognitive)

Conceptual Overview



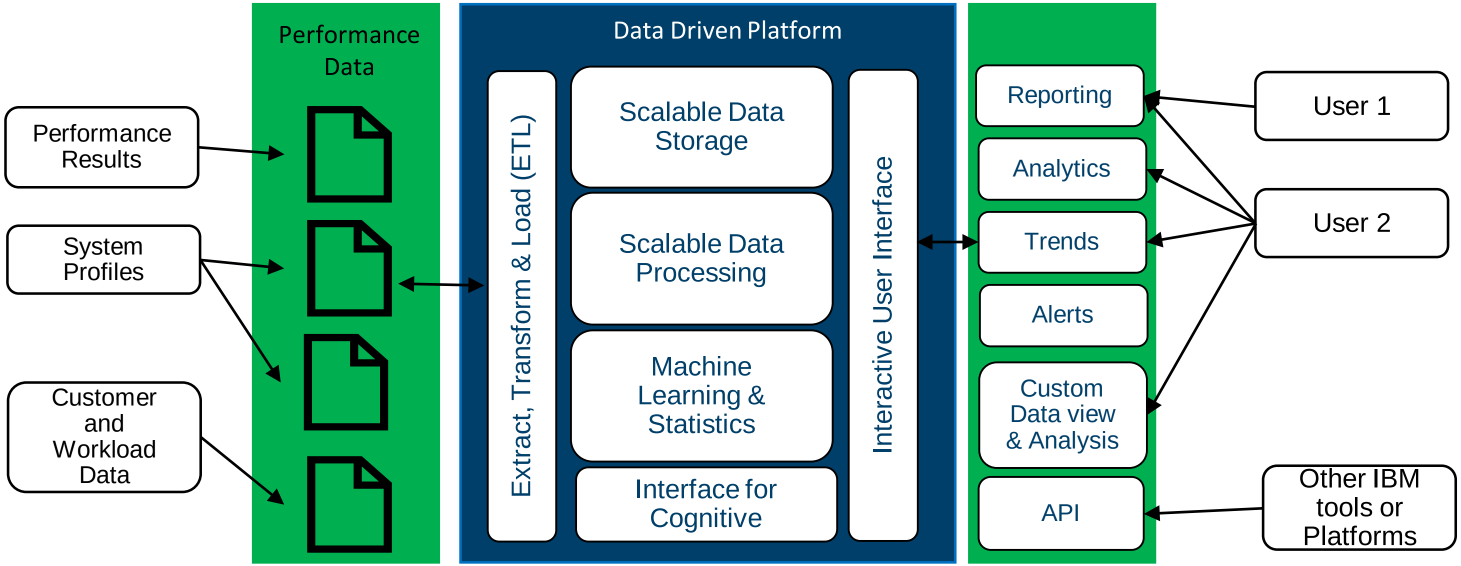
Results Presentation Layer

Performance Analysis dashboards

Tuning profiles classification and selection

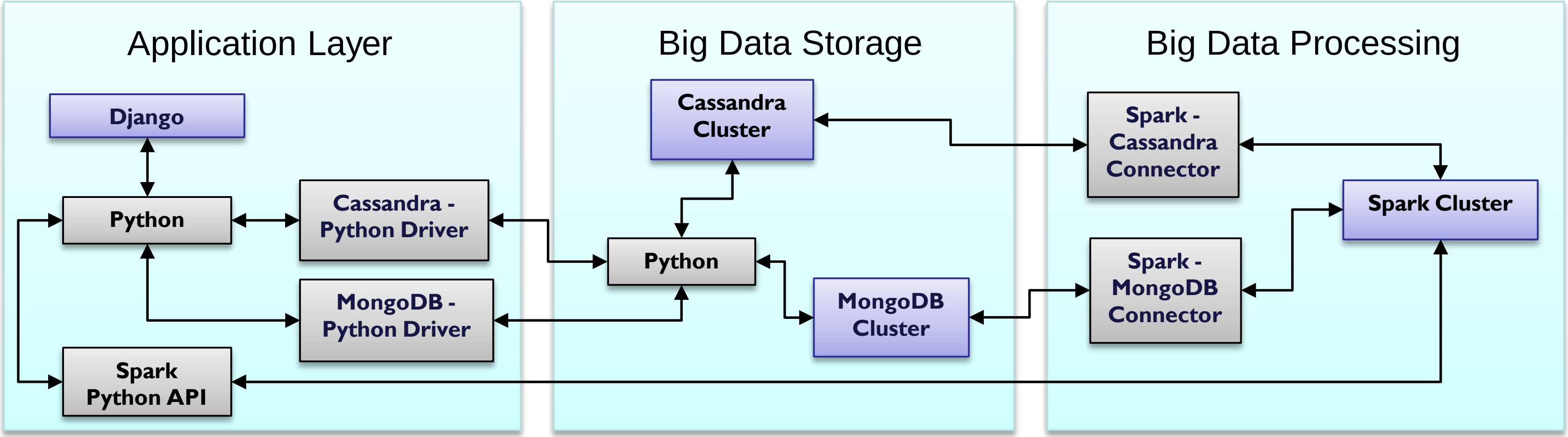
A Data Driven Performance Platform

Architecture Overview



A Data Driven Performance Platform

Deployment Overview



A Data Driven Performance Platform

Benefits

- Flexible, extensible, efficient, and scalable framework to store, query, and process performance data.
- Support historical query, analysis, and view of the results.
- Extend to not only store regression results but also other performance story results as well as new storage system evaluation results.
- Faster and centralized data availability.
- Custom data analysis to produce deeper insights about performance results and correlate them with customer, workload & systems configuration data.
- Use of Machine Learning to enhance the data utilization in configuration and tuning processes.

A ML-based configuration profiles repository

Example Use Case

“Selecting the closest tuning configuration profile according to system characteristics”.

Scenario:

Performance evaluations (for e.g., performance evaluation of new file system features or enhancements or workload, during regression tests on performance clusters), results in generation of the following data:

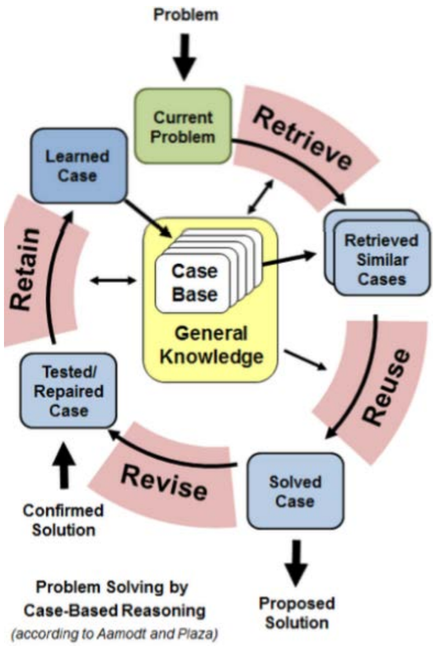
--Description of the overall solution (node CPU/memory, network, storage, disk-type, OS software etc).
--Description of the tested workload (DB, SWB, VDI, Metadata intensive, I/O access pattern).
--Full description of the tuned file system configuration.
--Observed Performance Results and conclusions.

Architecture + **Workload** + **File System Configuration** + **Performance Data** = **Environment Profile**

A ML-based configuration profiles repository

Case Based Reasoning

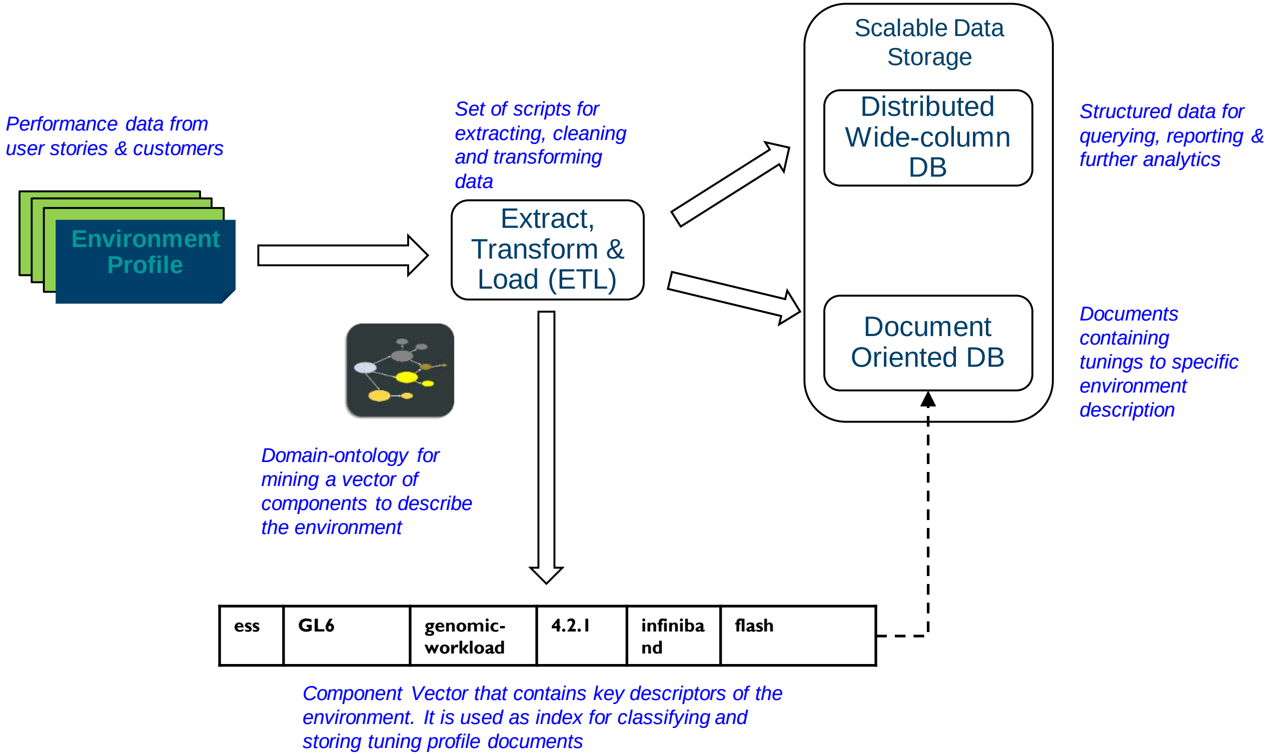
- An approach to model what humans think.
- An approach to build intelligent systems.
- Sub-discipline of Artificial Intelligence.
- Belongs to Machine Learning Methods.



- Recall **similar** experiences (made in the past) from memory.
- Reuse that experience in the context of the new situation (**reuse it partially, completely or modified**).
- New experience obtained this way is **stored to memory again**.

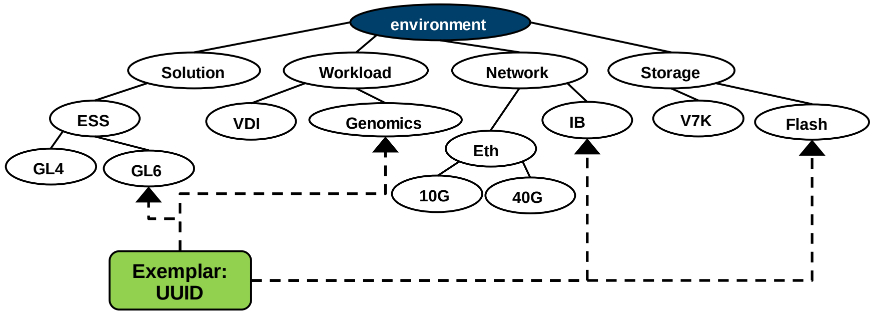
Classifying & recovering configuration profiles

Step 1: Using the ETL module for extracting and structuring the data

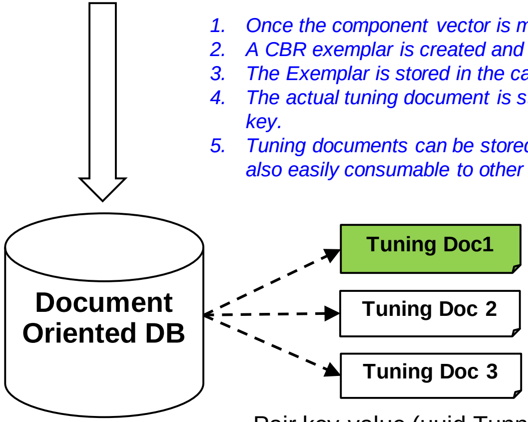


Classifying & recovering configuration profiles

Step 2: Storing the configuration profile and using ML (CBR) to assign indexes



1. *Once the component vector is mined, the descriptors are used to classify the new case.*
2. *A CBR exemplar is created and a UUID is assigned.*
3. *The Exemplar is stored in the case base.*
4. *The actual tuning document is stored in a document oriented DB using the Exemplar UUID as key.*
5. *Tuning documents can be stored as Json format in such way they are human readable and also easily consumable to other IBM tools.*



Pair key-value (uuid,Tunning Doc)

NOTE: *CBR is a ML technique which uses solutions of similar past cases to solve new problems. In this context, an exemplar is just a case in the case base.*

Classifying & recovering configuration profiles

Snippet of the Ontology using OWL

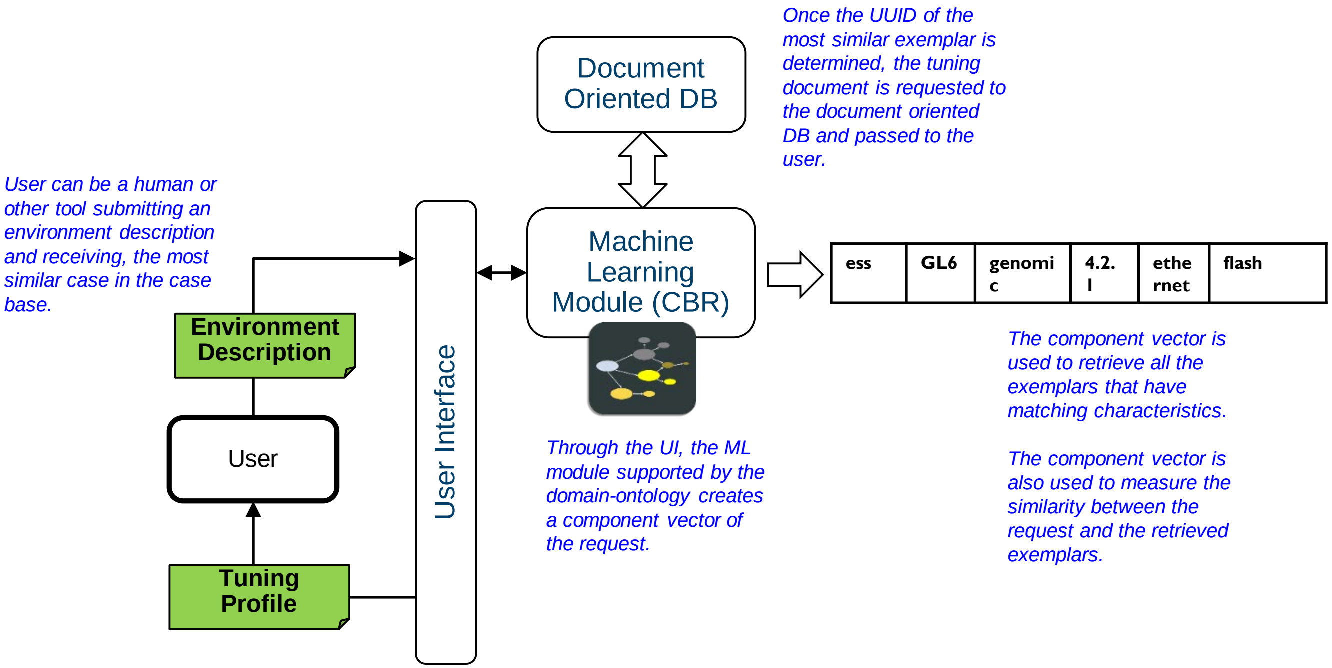
```
<!--
 http://www.ddp.com/ontologies/ss/environment.owl#InfinibandNetwork
-->
<owl:Class rdf:about="http://www.ddp.com /ontologies/ss/environment.owl#InfinibandNetwork">
   <rdfs:subClassOf rdf:resource="http:// www.ddp.com /ontologies/ss/environment.owl#ComputingNetwork"/>
   <rdfs:label xml:lang="en">IB</rdfs:label>
   <rdfs:label xml:lang="en">IB Network</rdfs:label>
   <skos:prefLabel xml:lang="en">InfiniBand</skos:prefLabel>
</owl:Class>
```

Snippet of the cases insertion in MongoDB

```
examplar = {"_id": "ess|gl6|genomicworkload|gpfs|5.0.1|infiniband|flash",
            "mmfsadm_dump_config":{"gpfs_parameters": {  "aclGarbageCollectorDutyCycle": 0,
                                                          "aclHashSpaceSize": 2000,
                                                          "afmAsyncDelay": 15,
                                                          "afmAsyncOpWaitTimeout": 300,
                                                            …
                                                        }
                                  }
            }
```

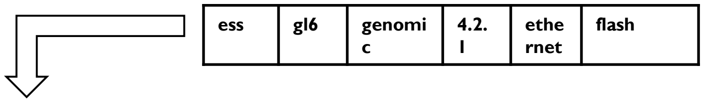
Classifying & recovering configuration profiles

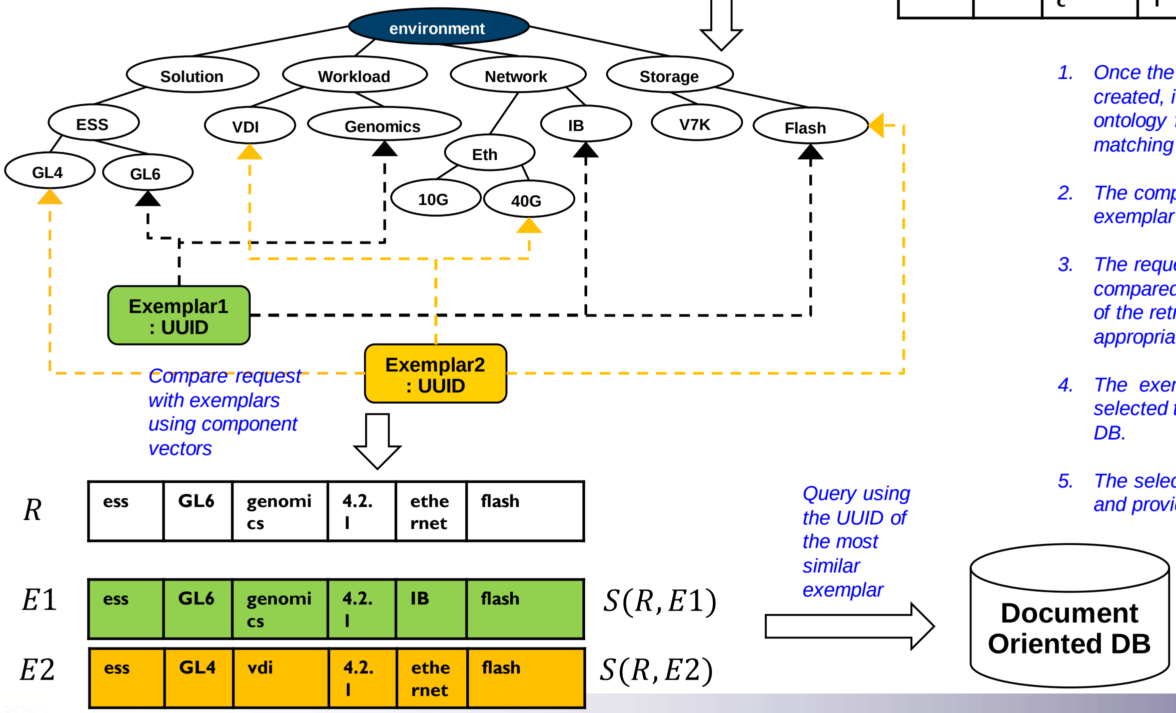
Step 3: Requesting the closest configuration profile based on the environment description.



Classifying & recovering configuration profiles

Step 4: Retrieving the closest configuration profile based on the environment description.





1. *Once the requested component vector is created, it needs to be parsed through the ontology for retrieving exemplars with matching characteristics.*
2. *The component vectors for each retrieved exemplar is built.*
3. *The requested component vector (R) is compared against the component vectors of the retrieved exemplars (En) using an appropriate similarity function S(R, En).*
4. *The exemplar with closest match is selected to query the Document Oriented DB.*
5. *The selected tuning document is retrieved and provided to the user.*

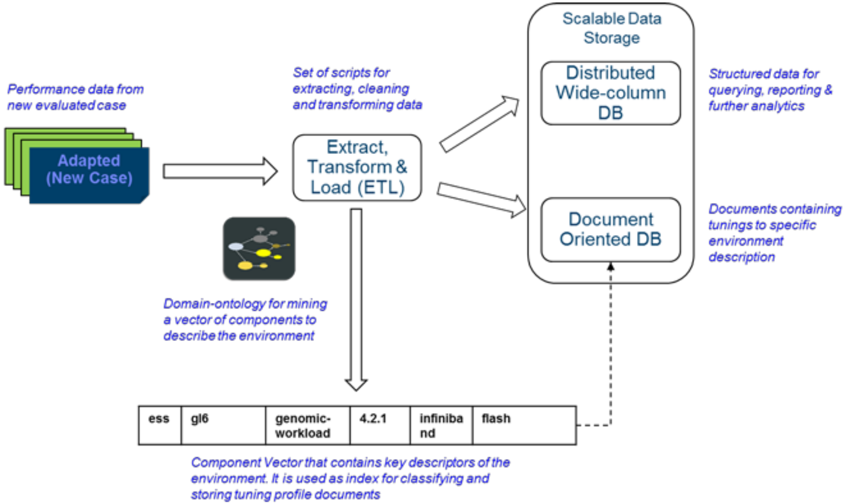
SDC 18

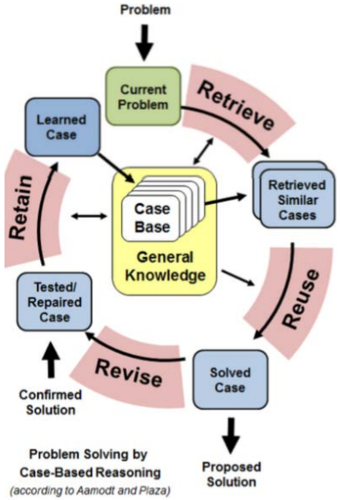
Classifying & recovering configuration profiles

Step 5: Evaluation and Adaptation of New Cases.

Once the tuning configuration of the most similar case is retrieved, it needs to be evaluated in the new system environment and adapted accordingly.

The adaptation of the tuning is then submitted to the proposed platform as a new case. Thus, this completes the CBR four-step process.





Classifying & recovering configuration profiles

Benefits of the Example Use Case

- Using a domain ontology as structure for classification allows to eliminate terminology ambiguity using synonyms and other relationships between the terms. For example: IB can be described as InfiniBand or Infini-Band.

- With more evaluated cases, the system develops expertise (similar to humans).

- The tuning profile is not just a document, but aggregation of performance data, workload, and system characteristics linked through appropriate data structures in the DB.

- The proposed architecture allows the interaction not only with humans but also with other tools aimed automating the entire configuration process.

Classifying & recovering configuration profiles

Related Successful Use Cases

Martin, Andreas, et al. "**A viewpoint-based case-based reasoning approach utilizing an enterprise architecture ontology for experience management.**" *Enterprise Information Systems* 11.4 (2017): 551-575.

De Renzis, Alan, et al. "**Case-based reasoning for web service discovery and selection.**" *Electronic Notes in Theoretical Computer Science* 321 (2016): 89-112.

Troiano, Luigi, Alfredo Vaccaro, and Maria Carmela Vitelli. "**On-line smart grids optimization by case-based reasoning on big data.**" *Environmental, Energy, and Structural Monitoring Systems (EESMS), 2016 IEEE Workshop on*. IEEE, 2016.

Dorn, Jürgen. "**Sharing Project Experience through Case-based Reasoning.**" *Procedia Computer Science* 99 (2016): 4-14.

Fragoso Diaz, Olivia G., Rene Santaolaya Salgado, and Ismael Solis Moreno. "**Using case-based reasoning for improving precision and recall in web services selection**." *International Journal of Web and Grid Services* 2.3 (2006): 306-330.

Ongoing Work

- We are in development, but need to evaluate the platform.
- Explore more similarity functions.
- Explore Deep Learning for the revising phase of CBR.
- Take this approach to different environments and significantly populate the knowledge base.
- Integrate the knowledge base with other approaches that are looking for the best configuration profiles.

Comments & Questions

Ismael Solís Moreno
ismael@mx1.ibm.com
https://www.linkedin.com/in/ismaelsm/