A Control of a mono and a multi scale measurement of a grid

Imene Elloumi elloumi@irit.fr
S. Ravelomanana raveloma@irit.fr
Manel Jelliti jelliti@irit.fr
Michelle Sibilla Sibilla@irit.fr
Thierry Desprats desprats@irit.fr
Outline

Motivation

Monitoring Pattern
- Control of a mono scale measurement
- Control of a multi-scale measurement

Illustration

Results and Future Work
**Context: Grid**

- A grid is a virtual infrastructure which consists of distributed and heterogeneous resources.
- Complex, composite and interdependent system.

<table>
<thead>
<tr>
<th>Knowledge that never changes</th>
<th>Knowledge that changes slowly</th>
<th>Knowledge that rapidly changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Initial Speed of processors, the memory amount available, …</td>
<td>Operating or not of the machine: on / off</td>
<td>Speed and Load CPU, performance of network: bandwidth, latency, jitter, …</td>
</tr>
</tbody>
</table>

$\Rightarrow$ QoS !
Problems

1. Need to mask heterogeneity of the format of management information and important number of metrics.

2. Among same limits of monitoring tools:
   - Lack of aggregation of information for more relevance at all levels.
   - A posteriori analysis of monitoring information.

3. Need for an inductive expertise.

4. No accurate diagnostic in real time.

5. Absence of preventive monitoring.
Towards to guarantee of the availability of grid

**Objective**

Towards to guarantee of the availability of grid

**Autonomy**

Control

Optimize

Monitor
(Observe+Measure)

Monitoring tools

Analyse
(Diagnostic)

Inductive Monitoring of Network in Real Time

Monitor / Analyse
Proposed Solution
An unified and standard approach based on:
1- Common Information Model CIM/DMTF + OGF partner
2- UML State Chart Diagram.

MOTIVATION
MONITORING PATTERN
ILLUSTRATION
RESULTS AND FUTURE WORK

3- Interesting observation pattern/DMTF that can cover one or an aggregation of observable property:

- Specifying the change of one critical state to other.
- Control and automate this observation pattern.

Sensor
+ SensorType : uint16 (enum)
+ PossibleStates : string[]
+ CurrentState : string
+ OtherSensorTypeDescription : string

MultiSensorType
- CollectionOfSensors
- MultiStateSensor
  - MultiSensorType : uint32
  - CurrentState : string
  - MaxNumberOfSensor : uint32

- Static Modelling
- Behavior Modelling

Network
System
Node
Of the grid
Routers, Switches and Links
Service
Applications deployed on the grid
Modeling the thresholds for a property observed:

**Observed Element** => **ManagedSystemElement**

**Sensor** => **NumericSensor**

**Controlled Property to set in the "Monitor" dependency**

**MultiSensor** = **MultiStateSensor**

**Dependency(Core)**
- Antecedent: ref ManagedElement
- Dependent: ref ManagedElement

**LogicalPort**
- Speed: uint64
- MaxSpeed: uint64

**Processing**
- LoadPercentage: uint16
- CurrentClockSpeed: uint16
- CPUStatus: uint16

**ManagedElement**
+ (see core model)

**LogicalElement**
+ (see core model)

**EnabledLogicalElement**
+ (see core model)

**LogicalDevice**
+ (see core model)

**Antecedent**:
- ref Sensor

**Dependent**:
- ref LogicalDevice

**PropertyName**:
- String

**PropertyType**:
- uint16 (enum)

**Monitor**
- CollectionOfSensors

**LogicPort**
- AssociatedSensor

**Sensor**
- AssociatedSensor

**ManagedSystemElement**
+ OperationStatus
- Status

**Sensor**
- BaseUnits: uint16
- CurrentReading: Sint32
- MinThresholdNoCritical: Sint32
- MinThresholdCritical: Sint32
- MaxThresholdNoCritical: Sint32
- MaxThresholdCritical: Sint32
- RestoreDefaultThresholds

**NumericSensor**
- BaseUnits: uint16
- CurrentReading: Sint32
- MinThresholdNoCritical: Sint32
- MinThresholdCritical: Sint32
- MaxThresholdNoCritical: Sint32
- MaxThresholdCritical: Sint32
- RestoreDefaultThresholds

**MultiStateSensor**
- MultiSensorType: uint32
- CurrentState: string
- MaxNumberOfSensor: uint32

**Dependency**
- Antecedent: ref ManagedElement
- Dependent: ref ManagedElement

**Antecedent**:
- ref NumericSensor

**Dependent**:
- ref LogicalDevice

**Processor**
- Speed: uint64
- MaxSpeed: uint64

**LogicalPort**
- AssociatedSensor

**AssociatedSensor**
- Antecedent: ref Sensor
- Dependent: ref LogicalDevice

**Monitor**
- CollectionOfSensors

**Sensor**
- AssociatedSensor

**ManagedSystemElement**
+ OperationStatus
- Status
Control of a mono scale measurement

Thresholds set for each propriety (Bandwidth, LoadCPU, SpeedCPU)

Administrator

MOTIVATION
MONITORING PATTERN
ILLUSTRATION
RESULTS AND FUTURE WORK
Control of a multi-scale measurement

Aggregation of management Information

Sensor
- SensorTypenum
- PossibleStatestring
- CurrentStatestring
- OtherSensorTypeDescribing

CollectionOfSensors

NumericSensor
- BaseUnitsenum
- CurrentReadingint
- MinThresholdNoCriticalint
- MaxThresholdNoCriticalint
- MinThresholdCriticalint
- MaxThresholdCriticalint

MultiStateSensor
- MultiSensorTypesenum
- CurrentStatestring
- MaxNumberOfSensorint

Need of Quantification
A Quantification of the criticality of states

Example

\[ N_1 = f_1(x)/F(x)=((\text{[number of sensor in a NoCritical state]}*10)\times100)/((\text{[number of sensor in a NoCritical state]}*10)+(\text{[number of sensor in a DegradeNoCritical state]}*15)+(\text{[number of sensor in a Critical state]}*20)+(\text{[number of sensor in a DegradeCritical state]}*25)+(\text{[number of sensor in a Fatal state]}*30)) \]
Extract of the textual description of the StateChart diagram example

MOF (Management Object Format) notation of DMTF

statemachine IRIT_NumericSensorMachine CIM_NumericSensor{
    state NoCritical{
        transition ("DegradedNoCritical")
        {
            on signal_event( CIM_InstanceModification ), condition Condition1{
                action1
            };
        }
    }
}...
Condition 1

expressed in CQL (CIM Query Language) conditionnal language

$Instance$ //the instance to which is associated statechart diagram

SELECT* // Specification of the evenement
FROM CIM_IndicationModification IM,
((SELECT
(PropertyName
FROM Monitor m
WHERE Antecedent = $Instance$
)PN
FROM LogicalDevice LD, Monitor m
WHERE m.Antecedent=$instance$
AND m.Dependant=LD
) VALUE,
// Specification of the condition from the NoCritical state: C1
(SELECT MinThresholdNoCritical
FROM NumericSensor
WHERE
$Instance$.System.CreationClassName=NumericSensor.SystemCreationClassName
AND
$Instance$.SystemName= NumericSensor.SystemName
AND
$Instance$.DeviceID= NumericSensor.DeviceID
) X1
WHERE IM.SourceInstance.SCCN=X1.SCCN
AND IM.SourceInstance.SN=X1.SN
AND IM.SourceInstance.DID=X1.DID
AND VALUE<C1.MinThresholdNoCritical

(SELSELECT SystemCreationClassName SCCN,
   SystemName SN, DeviceID DID
FROM LogicalDevice LD, Monitor m
WHERE m.Antecedent=$Instance$
AND m.Dependant=LD
)X1
WHERE IM.SourceInstance.SCCN=X1.SCCN
AND IM.SourceInstance.SN=X1.SN
AND IM.SourceInstance.DID=X1.DID
AND VALUE<C1.MinThresholdNoCritical

A pattern for multi-dimensional monitoring application example (Figure table III. MULTI-DIMENSIONAL STATE)

Hypothesis:
And a=10, B=15, C=20, D=25, E=30.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>MSS (Bandwidth)</th>
<th>MSS (Speed CPU)</th>
<th>MSS (Load CPU)</th>
<th>MSS (MMS Load and MMS Speed CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fi(x)=</td>
<td>f1(x)=120</td>
<td>f1(x)=70</td>
<td>f1(x)=100</td>
<td>f1(x)=0</td>
</tr>
<tr>
<td></td>
<td>f2(x)=405</td>
<td>f2(x)=165</td>
<td>f2(x)=105</td>
<td>f2(x)=0</td>
</tr>
<tr>
<td></td>
<td>f3(x)=640</td>
<td>f3(x)=320</td>
<td>f3(x)=280</td>
<td>f3(x)=0</td>
</tr>
<tr>
<td></td>
<td>f4(x)=375</td>
<td>f4(x)=75</td>
<td>f4(x)=375</td>
<td>f4(x)=1</td>
</tr>
<tr>
<td></td>
<td>f5(x)=420</td>
<td>f5(x)=390</td>
<td>f5(x)=120</td>
<td>f5(x)=1</td>
</tr>
<tr>
<td>F(x)=[f1(x)+f2(x)+f3(x)+f4(x)+f5(x)]/100</td>
<td>19.6</td>
<td>10.2</td>
<td>9.8</td>
<td>0.02</td>
</tr>
<tr>
<td>Valuation</td>
<td>N1&lt;N4&lt;N2&lt;N5=N3</td>
<td>N4&lt;N1&lt;N5=N3</td>
<td>N1&lt;N5&lt;N3=N4</td>
<td>N4=N1&lt;N4=N5</td>
</tr>
</tbody>
</table>

Multi-State Sensor

<table>
<thead>
<tr>
<th>Sensor</th>
<th>N2&gt;Ni, (i!=2) Then his State is « Degraded No-Critical »</th>
<th>N5&gt;Ni, (i!=5) Then his State is « Fatal »</th>
<th>N4&gt;Ni, (i!=4) Then his State is « Degraded Critical »</th>
<th>N5=N4 Then his State is « Fatal »</th>
</tr>
</thead>
</table>

![Diagram of Multi-State Sensor](image-url)
Contributions

- A standard model (CIM).
- Generic observation pattern (for mono and multi scale measurement).
- Control of transition from one state to another.
- Precision and personalization of control.
- Diagnostic of a component monitored.
- The fine scale with a same granularity that is instantiated for all properties monitored.
- Monitor of heterogeneous source.
Alarms release problem
Future Work

- Refining the analysis for more precision.
- Introduce the notion of time.
- Identify of a critical behavior: analysis improved of behavior in time + regular diagnostic of behavior → statistical analysis → detection of anomaly.
Future Work

- Implementing the solution in Openpegasus.
- Work in other metric: jitter and latency.
- Improve the quantification which must be adjustable (for example: add a class for this).
- Integrate the solution into the autonomic loop.

![Control Flow Diagram]

- Analyze + Diagnostic
- Observe + Measure
- Monitoring tools
- Control
- Optimize
Thanks for your Listening...

QUESTIONS