PoC #14: Intent-based Cloud Management
Progress Update

PoC host contact/Rapporteur:
NTT: Chao Wu

PoC member contacts/Co-Rapporteurs:
Intel: Emma Collins, Haining Wang, Chris Cavigioli
Intracom Telecom: Nikos Anastopoulos
NTT-AT: Takaaki Tanaka
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMF</td>
<td>Access and Mobility Management Function</td>
</tr>
<tr>
<td>LLC</td>
<td>Last Level Cache</td>
</tr>
<tr>
<td>NFV</td>
<td>Network Function Virtualisation</td>
</tr>
<tr>
<td>NUMA</td>
<td>Non-Uniform Memory Access</td>
</tr>
<tr>
<td>SLO</td>
<td>Service Level Objective</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SMF</td>
<td>Session Management Function</td>
</tr>
<tr>
<td>UPF</td>
<td>User Plane Function</td>
</tr>
<tr>
<td>VDI</td>
<td>Virtual Desk Infrastructure</td>
</tr>
</tbody>
</table>
PoC goal

**Intent**
(Functional and non-functional requirements)

- Security level
- Reliability level
- Performance
- Etc.

**Current approaches**

- Cloud consultant approach
  - High cost for cloud provider

- Self-service approach
  - Cloud skills demanded for cloud user

**Cloud Provider**

**Resource Design**

- Service Function Chain
- **Computation resource amount** (number of vCPU, etc.)

<table>
<thead>
<tr>
<th>FW</th>
<th>WS</th>
<th>DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCPU4,me</td>
<td>vCPU2,me</td>
<td>vCPU1,me</td>
</tr>
<tr>
<td>m4G, IO.</td>
<td>m4G, IO.</td>
<td>m4G, IO.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VM</th>
<th>VM</th>
<th>VM</th>
</tr>
</thead>
</table>

Cloud Infra.

FW: Firewall  WS: Web Server  DB: Database

**IBCM: autonomize the cloud resource decision**
PoC Abstract

**PoC Project Name**: Intent-based Cloud Management (IBCM)

**Short Description**: This PoC will provide an Intent-Based Cloud Management (IBCM) solution that assists the cloud provider with decision making about cloud computing resources, to meet the cloud performance goal, i.e. the intent.

In the PoC, we will demonstrate abstracting knowledge (building AI models) from cloud telemetry data, and making decisions of necessary cloud computing resources that meets the cloud performance goal using the knowledge (the models). Consequently, reduction of OPEX including the human resource cost, time cost and cloud resource cost can be expected by by the IBCM.
PoC Architecture
PoC user story

UC #1 Intent-based Cloud Management for VDI service

UC #2 Intent-based Cloud Management for NFV workloads
UC #1 Intent-based Cloud Management for VDI service

Preconditions: VDI users conduct their daily work in the virtual desktop instances provided by the VDI service.
Objective: The VDI service operator intends to maintain VDI users’ QoE by keeping the performance goal (i.e. the intent) satisfied, thus deciding the appropriate number of instances to be allocated to each host to meet the intent is necessary. The IBCM framework assists the operator with the resource decision.
UC #1 Intent-based Cloud Management for VDI service

**Step 0:** Collect the VDI performance log data using platform telemetry and build the IBCM model.

**Step 1:** The VDI operator specifies the intent through the GUI.

**Step 2:** IBCM checks if the current resource configuration meets the intent, if not, IBCM calculates the number of instances to be allocated to the host that meet the intent as well as the expected performance. The result is fed back to the operator for confirmation. The decision is transformed into machine-readable resource orchestration template and handed to VDI resource management system for implementation.

**User’s performance requirement (intent):**
- What is the maximum number of VMs for one host to meet the following requirements?
  - **host CPU ready time less than 100ms**
  - **Disk I/O latency less than 10ms**
  - **User input response time less than 1s**

**Current resource configurations:**
- Expected CPU ready time, Disk I/O latency and user input latency

**Recommended resource configurations:**
- Expected CPU ready time, Disk I/O latency and user input latency

**If current resource configuration cannot meet the intent:**
**UC #2 Intent-based Cloud Management for NFV workloads**

As a mobile core network operator, I need to find ways to easily & safely collocate NFV workloads on NFVi servers at the core or at the edge, so that server density gets increased, energy consumption gets minimized, while the workloads’ SLOs (provisioned as “intents”) are always maintained despite any dynamic change.

To do this, I need to leverage modern server technologies for efficient hardware resource partitioning (LLC, memory B/W, power) and isolation, combined with AI/ML techniques to accurately apply resource allocations at the right amount and time (no more than needed, and before SLA violations occur).

I know that I am successful when:
- the declared SLOs for high-priority workloads are always met, even when dynamic changes occur (e.g. traffic variation, workload arrival/departure)
- the overall power consumption gets reduced, as compared to the best possible deployment scheme that would not use any resource partitioning feature (either because servers can be entirely evacuated, or because they transition from symmetric to asymmetric power configurations)
- the overall time needed to discover optimal resource decisions gets reduced, as compared to the best possible manual/semi-manual approach for the same purpose
The operator offers two low-latency gaming services to its customers backed by differentiated 5G slices: a *premium* slice (ultra-low latency), and a *normal* slice (low latency). This essentially translates to two sets of 5G functions (e.g. UPF, SMF, AMF) with differentiated intents: e.g. premium UPF < 0.07 msec, normal UPF < 0.3 msec.

- **“AS IS” state:**
  - without resource partitioning technologies in place, the operator would isolate each UPF instance to its own NUMA node, reserving upfront any cores left idle in order to avoid contention that would put SLAs under risk. At large scale, a larger number of NFV servers would be needed to host many UPF instances.
  - with resource partitioning techniques, the operator would need much time and expertise to discover colocated placements that would reduce the total number of servers needed. Even in that case, however, he should experiment assuming the worst-case scenario for each VNF (i.e. max expected traffic), thus missing opportunities for even denser consolidation in quiet periods.
UC #2 Intent-based Cloud Management for NFV workloads
Differentiated 5G online gaming services

“TO BE” state

Step 1: user input
- Performance-level intents (SLOs)
  - premium UPF latency < 0.07 msec
  - normal UPF latency < 0.3 msec
- Training data
  - diurnal load/traffic patterns, as expected in production

Step 2: train IBCM on intents, in a staging environment
- IBCM tests many different resource configs for all colocated workloads, and for different traffic levels, in order to discover the configs that deliver the intended SLOs with minimal HW resource footprint
  - actions:
    - allocate LLC partition for a workload (Intel RDT)
    - set memory B/W for a workload (Intel RDT)
    - set base frequency for a workload (Intel SST)
- On every tested config, IBCM gets feedback relevant to the intents using telemetry from various levels (platform, workloads)

Step 3: use trained IBCM model in the production environment for local & real-time decision making
- The production environment needs to be identical to the staging environment, for best reproducibility
- Intents are encoded in the IBCM model, which is trained for the specific application(s) and for the specific traffic bounds provided by the user
- Different applications, intents, traffic bounds will typically require retraining
## PoC Milestones

<table>
<thead>
<tr>
<th>PoC Milestone</th>
<th>Stages/Milestone Description</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>P.S</td>
<td>PoC Project Start</td>
<td>June 2021, ENI #18</td>
</tr>
<tr>
<td>P.U</td>
<td>PoC user story</td>
<td>September 2021, ENI #19</td>
</tr>
<tr>
<td></td>
<td>NTT R&amp;D Forum 2021</td>
<td>November 2021</td>
</tr>
<tr>
<td>P.D1</td>
<td>PoC Demo</td>
<td>December 2021, ENI#20</td>
</tr>
<tr>
<td>P.C</td>
<td>PoC Contribution</td>
<td>March 2022</td>
</tr>
<tr>
<td>P.R</td>
<td>PoC Report</td>
<td>March 2022</td>
</tr>
<tr>
<td>P.E</td>
<td>PoC Project End</td>
<td>June 2022</td>
</tr>
</tbody>
</table>
Q: How much data/time is required? (UC#1)

Log data of at least one typical workday is necessary. (About 500 records of log data, 30 kinds of workload, resource and performance data)

It takes less than 10 minutes to train the model with one day’s log data. And the average inference time is less than 5 seconds.

More ideally, weekly collected log data is preferred to guarantee better precision.
Q: How much data/time is required? (UC #2)

Depends on the application complexity and number of actions allowed

Initially staging of the application is required (for now)
✓ Samples are gathered for different load/traffic levels for all available actions
✓ For simple applications at least 10 traffic levels are recommended

Such a process for one action dimension (e.g. LLC slicing) requires approx. 30 minutes and the input data are minimal (<10kB). The resulting model is approx. 10MB in size
List of contributors

Chao Wu (NTT)  Evangelos Angelou (Intracom Telecom)
Emma Collins (Intel)  Georgia Panoutsakopoulou (Intracom Telecom)
Haining Wang (Intel)  Loukas Gravias (Intracom Telecom)
Chris Cavigioli (Intel)  Victor Timofei (Intracom Telecom)
Anastopoulos Nikos (Intracom Telecom)  Georgios Hortatos (Intracom Telecom)
Shingo Horiuchi (NTT)  
Kenji Murase (NTT)  
Hiroaki Kikushima (NTT)  
Kenichi Tayama (NTT)  
Browne John (Intel)  
Macnamara, Chris (Intel)  