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

Acronyms

AMF	Access and Mobility Management Function
LLC	Last Level Cache
NFV	Network Function Virtualisation
NUMA	Non-Uniform Memory Access
SLO	Service Level Objective
SLA	Service Level Agreement
SMF	Session Management Function
UPF	User Plane Function
VDI	Virtual Desk Infrastructure

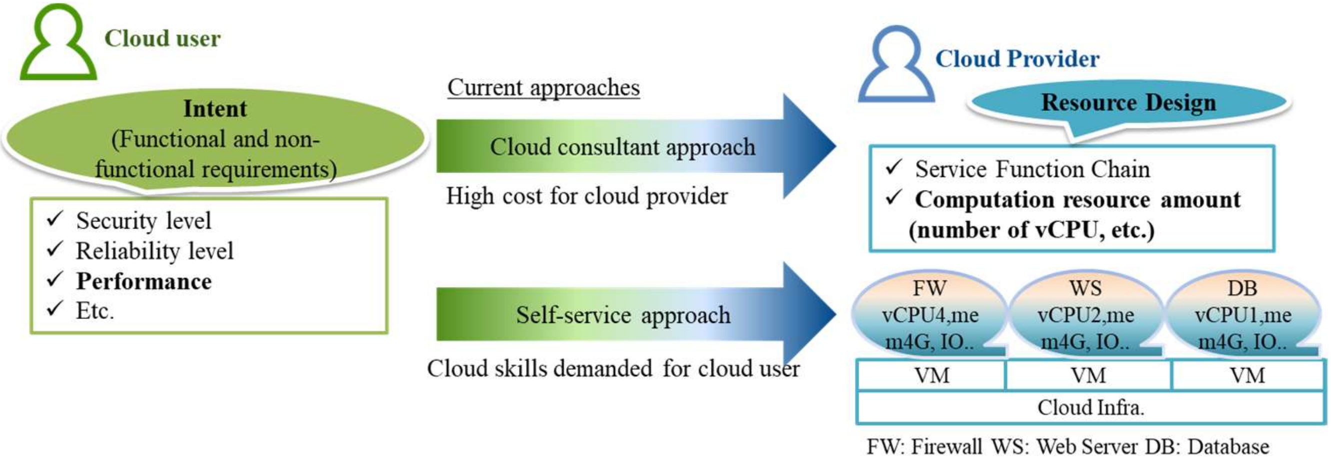
PoC milestones

PoC Milestone	Stages/Milestone description	Target Date
P.S	PoC Project Start	June 2021, ENI #18
P.U	PoC user story	September 2021, ENI #19
	NTT R&D Forum 2021	November 2021
P.D1	PoC Demo	December 2021, ENI#20
P.C	PoC Contribution	March 2022
P.R	PoC Report	March 2022
P.E	PoC Project End	June 2022

Current status

New milestone

PoC goal



IBCM: autonomize the cloud resource decision

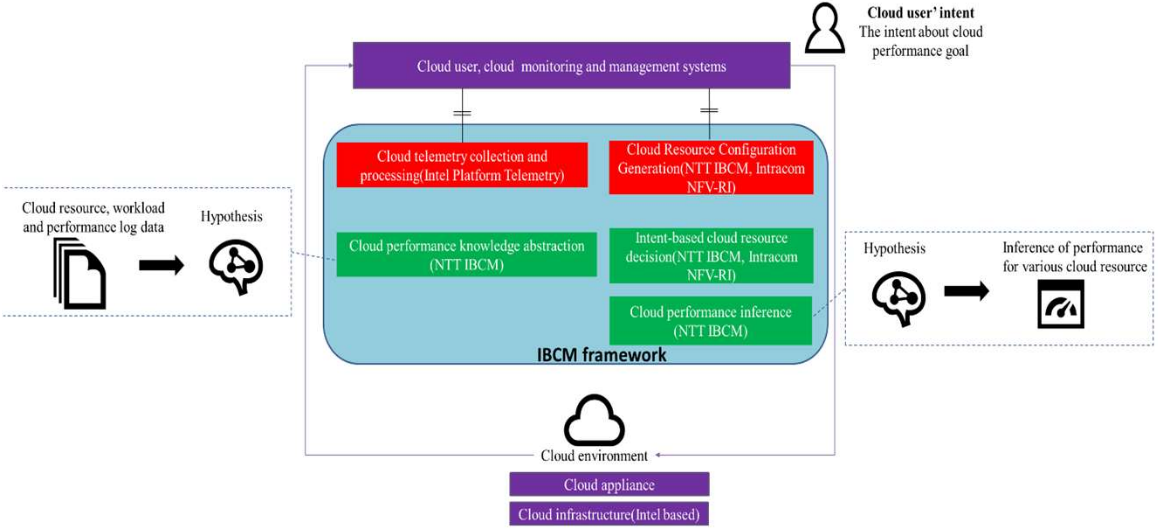
PoC Abstract

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 the IBCM.

PoC Architecture


Cloud user' intent
The intent about cloud performance goal
Cloud user, cloud monitoring and management systems
Cloud telemetry collection and processing(Intel Platform Telemetry)
Cloud Resource Configuration Generation(NTT IBCM, Intracom NFV-RI)
Cloud performance knowledge abstraction (NTT IBCM)
Intent-based cloud resource decision(NTT IBCM, Intracom NFV-RI)
Cloud performance inference (NTT IBCM)
IBCM framework
Cloud resource, workload and performance log data
Hypothesis
Hypothesis
Inference of performance for various cloud resource
Cloud environment
Cloud appliance
Cloud infrastructure(Intel based)

PoC user story

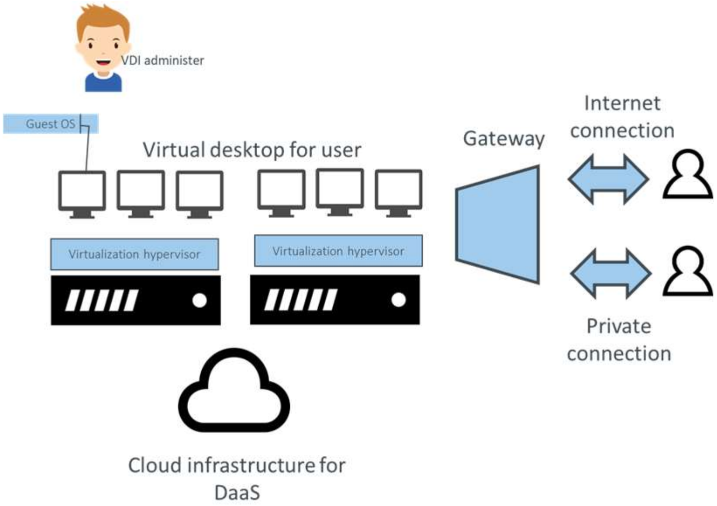
UC #1 Intent-based Cloud Management for VDI service

UC #2 Intent-based Cloud Management for NFV workloads

UC #1 Intent-based Cloud Management for VDI service

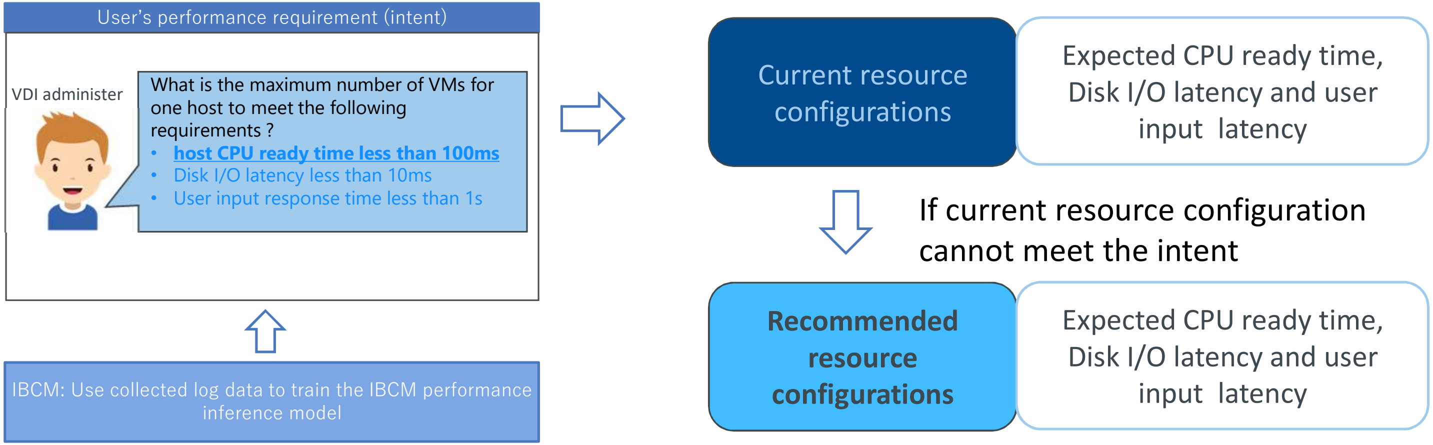
Preconditions: VDI users conduct their daily work in the virtual desktop instances provided by the VDI service.

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

Step0: Collect the VDI performance log data using platform telemetry and build the IBCM model.
Step1: The VDI operator specifies the intent through the GUI.
Step2: 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)

VDI administer

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

IBCM: Use collected log data to train the IBCM performance inference model

Current resource configurations

Expected CPU ready time, Disk I/O latency and user input latency

If current resource configuration cannot meet the intent

Recommended resource configurations

Expected CPU ready time, Disk I/O latency and user input latency

UC #2 Intent-based Cloud Management for NFV workloads

As a mobile core network operator
I need to find ways to easily & safely colocate 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 (provided 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

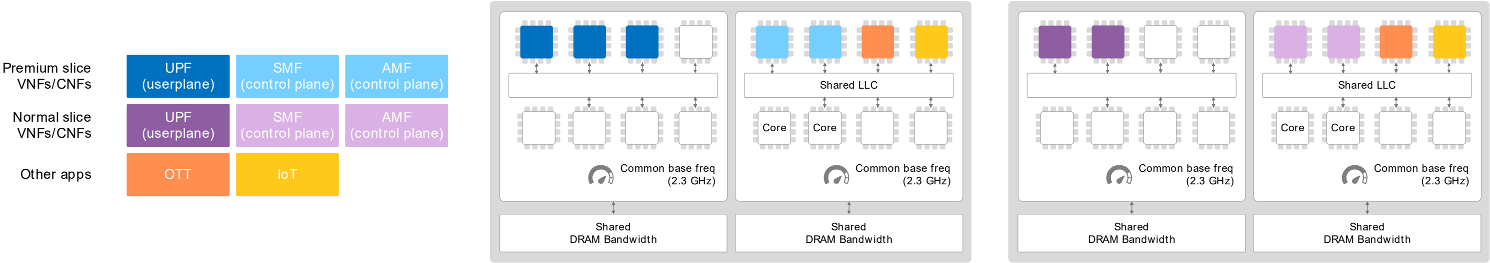
- the declared **SLOs** for high-priority workloads are always met, even when dynamic changes occur (e.g. traffic variation, workload arrival/departure)
- the overall **power consumption** gets reduced, as compared to the best possible deployment scheme that would not use any resource partitioning feature (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

UC #2 Intent-based Cloud Management for NFV workloads

Differentiated 5G online gaming services

- 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 5GC 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 NFVi 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

IBCM

Step 2: train IBCM on intents, in a staging environment

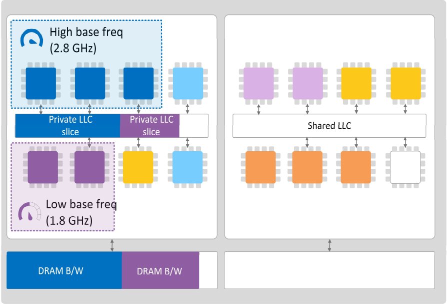
IBCM tests many different resource configs for all colocated workloads, and for different traffic levels, in order to discover the configs that deliver the intended SLOs with minimal HW resource footprint

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)

Premium slice VNFs/CNFs	UPF (userplane)	SMF (control plane)	AMF (control plane)
Normal slice VNFs/CNFs	UPF (userplane)	SMF (control plane)	AMF (control plane)
Other apps	OTT	IoT	



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

PoC milestones

PoC Milestone	Stages/Milestone description	Target Date
P.S	PoC Project Start	June 2021, ENI #18
P.U	PoC user story	September 2021, ENI #19
	NTT R&D Forum 2021	November 2021
P.D1	PoC Demo	December 2021, ENI#20
P.C	PoC Contribution	March 2022
P.R	PoC Report	March 2022
P.E	PoC Project End	June 2022

Current status

New milestone

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