Milestone #2

Data source

cluster-trace-microservices-v2022. The released traces contain the detailed runtime metrics of nearly twenty thousand microservices. They are collected from Alibaba production clusters of over ten thousand bare-metal nodes during 13 days in 2022.

Introduction of Trace Data

Node:

columns	Example Entry
timestamp	60000
nodeid	NODE_10632
cpu_utilization	0.266488095525847
memory_utilization	0.159064258887333

- timestamp: Timestamp of recorded metrics. The recording interval is the 60s (60 * 1000).
- nodeid: The specific id of BM node. It could be joined with nodeid in MS_Resource_Table.
- cpu_utilization: **Normalized** CPU utilization of BM node.
- memory_utilization: **Normalized** memory utilization of BM node.

MSResource:

columns	Example Entry
timestamp	180000
msname	MS_21881
msinstanceid	MS_21881_POD_0
nodeid	NODE_11517
cpu_utilization	0.21995999999530616
memory_utilization	0.833001454671224

- timestamp: Mentioned in Node. The recording interval is the 60s (60 * 1000).
- msname: The name of MS, to be joined with MSName in MS_MCR_RT_Table, and DM and UM in MS_CallGraph_Table. MSName only contains stateless services, as stateful services run in other dedicated clusters.
- msinstanceid: The specific container id of MS. An MS may have more than one container.
- nodeid: The specific BM node in which MSInstanceID runs.
- cpu_utilization: **Normalized** CPU utilization of MSInstanceID.
- memory_utilization: **Normalized** memory utilization of MSInstanceID.

MSCallGraph:

columns	Example Entry
timestamp	115352
traceid	T_11560863075
service	S_153587416
rpc_id	0.1
um	MS_58845
uminstanceid	MS_58845_POD_0
rpctype	rpc
interface	xOuy6-80Vt
dm	MS_71712

columns	Example Entry
dminstanceid	MS_71712_POD_244
rt	2.0

- timestamp: Mentioned in Node.
- traceid: Each call graph has a unique traceID.
- service: Online service id. A specific online service provides a function for users. For example, the online shopping application can provide multiple online services, including ordering, goods searching, delivering and so on.
- rpcid: Each call is identified by a unique rpcID, which contains the ID information of a pair of UM and DM. For example, rpcID 0.1.1 and 0.1.2 denote two calls that two different DMs are called by the same UM, which is the DM in the call with rpcID 0.1. Note that, the call via remote invocation is recorded twice with the same rpcID in the UM and DM independently.
- um: The name of UM.
- uminstanceid: The specific container id of um MS. An MS may have more than one container.
- rpctype: The communication paradigms. We record rpc_type as "DB" and "MC" for the calls via inter-process communication if DM is DB and MC respectively.
- interface: The interface of DM is called by UM. The calls via remote invocation or HTTP have the interface.
- dm: The name of DM.
- dminstanceid: The specific container id of dm MS. An MS may have more than one container.

Data Cleaning

The dataset is more than 2TB but many of the fields of the schemes it has are not necessary. For instance, in `MSCallGraph`, the `rpc_id`, `service`, and `interface` are redundant for our later analysis. I am using a MapReduce application to clean all the unnecessary fields that I hardcoded in it.

Data Profiling

As an initial step of data analysis, profiling includes collecting the distribution of usages of memory and CPU in each Node and as well for each Microservice in the whole time span. We are also interested in the hotspot within the cluster in a specific time interval. Both profilings are done by MapReduce applications: `NodeUsage`, `MSUsage`.

Note: Though `NodeUsage` and `MSUsage` are rather similar, I decoupled them for later modifications and better readability.

Data Snippet

A data snippet of `CallGraph` before cleaning:

```
timestamp,traceid,service,rpc_id,rpctype,um,uminstanceid,interface,dm,dminstanceid,rt
115352,T_11560863075,S_153587416,0.1,rpc,MS_58845,MS_58845_POD_0,xOuy6-80Vt,MS_71712,MS_71712_POD_244,2.0
86450,T_22121589080,S_1798353,0.1,rpc,MS_24094,MS_24094_POD_3317,4Db39LxDTC,MS_24094,MS_24094_POD_5168,5.0
86450,T_22121589080,S_1798353,0.1,rpc,MS_24094,MS_24094_POD_983,4Db39LxDTC,MS_24094,MS_24094_POD_5168,5.0
155753,T_22121586395,S_85905920,0.1,rpc,MS_24094,MS_24094_POD_5463,1oNt-EK5Lm,MS_67455,MS_67455_POD_222,9.0
90649,T_22825688569,S_160146479,0.1.1.1,http,MS_29868,MS_29868_POD_22,XQrDXTN_dB,UNKNOWN,UNKNOWN,18.0
90664,T_22825688569,S_160146479,0.1.1.1.1.2.1,rpc,MS_51514,MS_51514_POD_3,lgmp2NziFu,MS_10343,MS_10343_POD_5,0.0
167154,T_20213377810,S_93592909,0.1.1.1.26.3.3,mc,MS_11245,MS_11245_POD_284,DqhrtBes7F,MS_60125,MS_60125_POD_5,1.0
152182,T_23983760905,S_32048416,0.1.1,db,MS_20664,MS_20664_POD_366,q0Y1mhsdUg,MS_66701,MS_66701_POD_10,1.0
99052,T_23620532973,S_86474918,0.1.3,db,MS_44922,MS_44922_POD_13,MMggwBTupk:TDDL_QUERY,MS_4062,MS_4062_POD_0,1.0
```

After cleaning:

```
traceid,um,uminstanceid,dm,dminstanceid
T_10000007374,100841,T_10000007374,S_10694245,0,http,USER,MS_52059,MS_52059_POD_10
T_10000007374,100841,T_10000007374,S_10694245,0.1,rpc,UNAVAILABLE,MS_71843,MS_71843_POD_166
T_10000021144,114931,T_10000021144,S_40318420,0.1.1.1,http,MS_745,MS_2548,MS_2548_POD_8
T_10000021144,114929,T_10000021144,S_40318420,0.1,rpc,UNAVAILABLE,MS_3351,MS_3351_POD_860
T_10000021144,114930,T_10000021144,S_40318420,0.1.1,http,UNKNOWN,MS_745,MS_745_POD_1523
T_10000021144,114929,T_10000021144,S_40318420,0,http,USER,MS_52059,MS_52059_POD_11
T_10000082152,118157,T_10000082152,S_74444256,0.1.1.1,http,MS_745,MS_65689,MS_65689_POD_3
T_10000082152,118153,T_10000082152,S_74444256,0.1,rpc,UNAVAILABLE,MS_3351,MS_3351_POD_173
T_10000082152,118157,T_10000082152,S_74444256,0.1.1,http,UNKNOWN,MS_745,MS_745_POD_2141
```

Acknowledgment

All the data information are taken from the Alibaba's clusterdata repo.