

# machine\_configs

March 10, 2021

## 1 Machine configurations

This query returns all the distinct NCU/NMU configurations in the borg clusters, including how many machines ids match for any specific configuration.

Please note that for simplicity's sake the we are technically counting the number of ADD or UPDATE events for each configuration, and not the actual count of machines. Therefore a machine configuration may change over time and count twice or more.

```
[1]: import pandas as pd

# For pretty printing
from IPython.display import display

# Disables row ellipsis
pd.set_option('display.max_rows', 200)

[3]: # Load all machine event rows in a single DataFrame, and add a "cluster" column
      ↳to differentiate
      # between clusters
df = None
for l in "abcdefgh":
    dfl = pd.read_csv("~/google_2019/machine_events/" + l + "_machine_events.
↳csv")
    dfl["cluster"] = l
    if df is None:
        df = dfl
    else:
        df = pd.concat([df, dfl], axis=0)

# Filter only ADD or UPDATE events
df = df[(df.type==1) | (df.type==3)]

# P.S: ADD=1, REMOVE=2, UPDATE=3

df = df[["capacity.cpus", "capacity.memory", "cluster",
        "missing_data_reason", "machine_id"]]
```

```
[4]: # Checking if we need to deal with particular missing data
# No columns returned, so missing data can be safely ignored
df.groupby(by=["missing_data_reason"], dropna=False).count()
```

```
[4]:          capacity.cpus  capacity.memory  cluster  machine_id
missing_data_reason
NaN                523781            523781   532510     532510
```

```
[5]: def do_group_by(df):
# Exclude "cluster" column and perform group-by
dfg = df[df.columns.difference(['cluster'])]. \
    groupby(by=["capacity.cpus", "capacity.memory"],
            dropna=False).count()

# Compute relative number of machines
total_machines = dfg['machine_id'].sum()
dfg["machine_id_perc"] = dfg["machine_id"] * 100 / total_machines

# Sort descending
dfg = dfg.sort_values("machine_id_perc", ascending=False)

display(dfg)
```

```
[6]: # Generate machine configurations table per cluster + a global table

df = df[df.columns.difference(['missing_data_reason'])]

for l in "abcdefgh":
    print("\nFor cluster " + l + ":\n")
    do_group_by(df[df.cluster==l])

print("\n For all clusters:")
do_group_by(df)
```

For cluster a:

capacity.cpus	capacity.memory	machine_id	machine_id_perc
0.591797	0.333496	29487	34.758469
1.000000	0.500000	13440	15.842705
0.708984	0.333496	12495	14.728764
0.386719	0.333496	9057	10.676144
	0.166748	5265	6.206238
0.708984	0.666992	4608	5.431784
1.000000	1.000000	4446	5.240823
0.591797	0.166748	2484	2.928071

NaN	NaN	1377	1.623170
0.958984	0.500000	1143	1.347337
	1.000000	654	0.770917
1.000000	0.250000	366	0.431431
0.479492	0.250000	6	0.007073
0.708984	0.250000	6	0.007073

For cluster b:

		machine_id	machine_id_perc
capacity.cpus	capacity.memory		
0.591797	0.333496	16184	31.982926
1.000000	0.500000	9790	19.347061
0.708984	0.333496	8448	16.694992
0.958984	0.500000	5502	10.873088
0.708984	0.666992	3832	7.572823
1.000000	1.000000	2214	4.375321
0.591797	0.166748	2152	4.252796
0.386719	0.333496	816	1.612584
0.958984	1.000000	618	1.221296
0.591797	0.666992	500	0.988103
0.386719	0.166748	412	0.814197
NaN	NaN	134	0.264812

For cluster c:

		machine_id	machine_id_perc
capacity.cpus	capacity.memory		
0.259277	0.166748	15754	24.439204
0.386719	0.333496	11104	17.225652
0.591797	0.333496	10404	16.139741
0.958984	0.500000	6634	10.291334
1.000000	0.500000	5654	8.771059
0.386719	0.166748	3580	5.553660
0.708984	0.666992	2900	4.498774
1.000000	1.000000	2736	4.244361
	0.250000	2132	3.307375
NaN	NaN	1466	2.274208
0.958984	1.000000	766	1.188297
0.708984	0.333496	620	0.961807
0.958984	0.250000	600	0.930781
0.591797	0.166748	112	0.173746

For cluster d:

capacity.cpus	capacity.memory	machine_id	machine_id_perc
0.591797	0.333496	28394	45.288376
0.386719	0.333496	8402	13.401174
0.259277	0.166748	8020	12.791885
0.386719	0.166748	5806	9.260559
0.708984	0.666992	4380	6.986092
	0.333496	3924	6.258772
0.591797	0.166748	2548	4.064055
NaN	NaN	498	0.794309
0.259277	0.333496	426	0.679469
1.000000	0.500000	292	0.465739
0.591797	0.250000	4	0.006380
0.708984	0.500000	2	0.003190

For cluster e:

capacity.cpus	capacity.memory	machine_id	machine_id_perc
0.259277	0.166748	38452	48.202377
0.708984	0.333496	11786	14.774608
0.958984	0.500000	8646	10.838389
0.708984	0.666992	7606	9.534674
1.000000	0.500000	5586	7.002457
0.386719	0.166748	4470	5.603470
0.259277	0.333496	1268	1.589530
	0.083374	634	0.794765
NaN	NaN	536	0.671915
0.591797	0.333496	324	0.406158
1.000000	0.250000	268	0.335957
	1.000000	138	0.172993
0.500000	0.062500	54	0.067693
	0.250000	4	0.005014

For cluster f:

capacity.cpus	capacity.memory	machine_id	machine_id_perc
1.000000	0.500000	41340	66.396839
0.708984	0.333496	6878	11.046866
0.591797	0.333496	5564	8.936430
0.958984	0.500000	2172	3.488484
0.386719	0.166748	1544	2.479843
NaN	NaN	1432	2.299958
0.708984	0.666992	1244	1.998008
1.000000	0.250000	792	1.272044

0.958984	1.000000	536	0.860878
0.386719	0.333496	398	0.639234
1.000000	1.000000	344	0.552504
0.500000	0.250000	18	0.028910

For cluster g:

capacity.cpus	capacity.memory	machine_id	machine_id_perc
0.259277	0.166748	15852	22.892958
1.000000	0.500000	11808	17.052741
0.708984	0.333496	7968	11.507134
0.591797	0.333496	7830	11.307839
0.386719	0.166748	4690	6.773150
0.708984	0.666992	4258	6.149269
0.958984	0.500000	4196	6.059731
0.386719	0.333496	3864	5.580267
0.591797	0.166748	2606	3.763503
1.000000	0.250000	2100	3.032754
NaN	NaN	1566	2.261568
0.259277	0.333496	1330	1.920744
0.958984	1.000000	778	1.123563
1.000000	1.000000	378	0.545896
0.500000	0.250000	12	0.017330
0.479492	0.250000	6	0.008665
	0.500000	2	0.002888

For cluster h:

capacity.cpus	capacity.memory	machine_id	machine_id_perc
1.000000	0.500000	36324	61.946178
0.591797	0.333496	4826	8.230158
0.708984	0.333496	3682	6.279205
0.958984	0.500000	2858	4.873973
0.386719	0.333496	2596	4.427163
1.000000	1.000000	2030	3.461919
	0.250000	1892	3.226577
NaN	NaN	1720	2.933251
0.386719	0.166748	1244	2.121491
0.708984	0.666992	766	1.306320
0.591797	0.666992	500	0.852689
0.958984	1.000000	200	0.341076

For all clusters:

capacity.cpus	capacity.memory	machine_id	machine_id_perc
1.000000	0.500000	124234	23.329891
0.591797	0.333496	103013	19.344801
0.259277	0.166748	78078	14.662260
0.708984	0.333496	55801	10.478864
0.386719	0.333496	36237	6.804943
0.958984	0.500000	31151	5.849843
0.708984	0.666992	29594	5.557454
0.386719	0.166748	27011	5.072393
1.000000	1.000000	12286	2.307187
0.591797	0.166748	9902	1.859496
NaN	NaN	8729	1.639218
1.000000	0.250000	7550	1.417814
0.958984	1.000000	3552	0.667030
0.259277	0.333496	3024	0.567877
0.591797	0.666992	1000	0.187790
0.259277	0.083374	634	0.119059
0.958984	0.250000	600	0.112674
0.500000	0.062500	54	0.010141
	0.250000	34	0.006385
0.479492	0.250000	12	0.002253
0.708984	0.250000	6	0.001127
0.591797	0.250000	4	0.000751
0.708984	0.500000	2	0.000376
0.479492	0.500000	2	0.000376

[ ]: