## A. Workload classification and clustering

为了对节点进行分类，我们对各节点随时间变化的cpu和memory利用率进行了分析，并做出了节点cpu\_usage和memory\_usage随时间的变化曲线。根据这些曲线特征，我们将所有的节点分成10类。下面将对这10类节点进行详细的分析。

1. 不同类型节点的描述

A类节点：cpu\_usage/memory\_usage均属于正常波动。例1号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

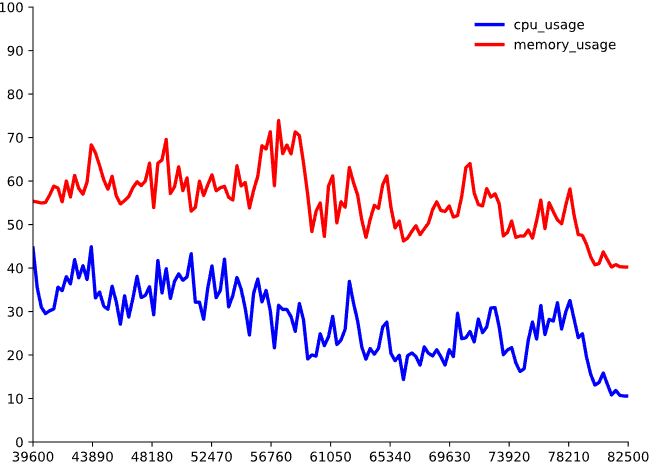


Figure 1 A类节点cpu\_usage/memory\_usage随时间的变化曲线

B类节点：cpu\_usage在52000s前波动较小。例37号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

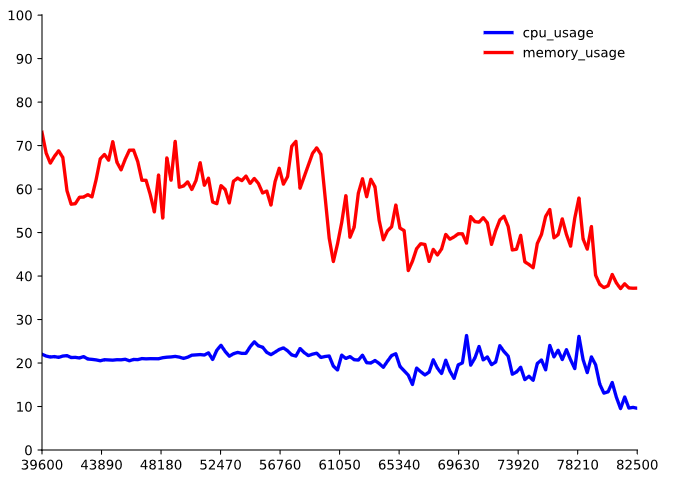


Figure 2 B类节点cpu\_usage/memory\_usage随时间的变化曲线

C类节点：cpu\_usage/memory\_usage在52000s后均波动较小。例401号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

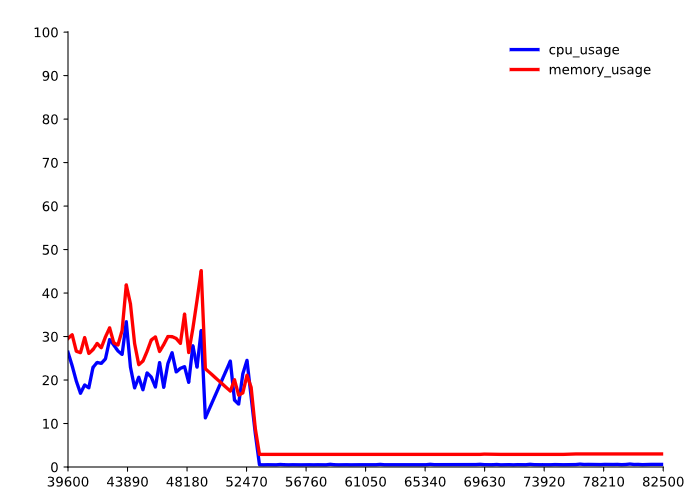


Figure 3 C类节点cpu\_usage/memory\_usage随时间的变化曲线

D类节点：cpu\_usage/memory\_usage数据缺失。例930号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

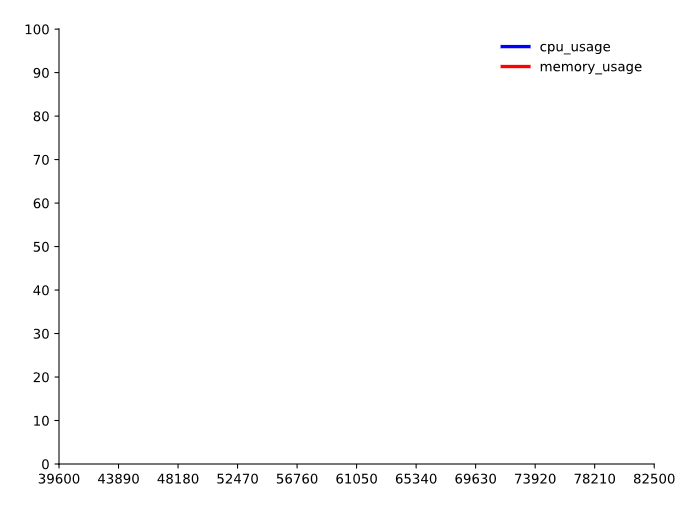


Figure 4 D类节点cpu\_usage/memory\_usage随时间的变化曲线

E类节点：cpu\_usage/memory\_usage全程无波动。例1075号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

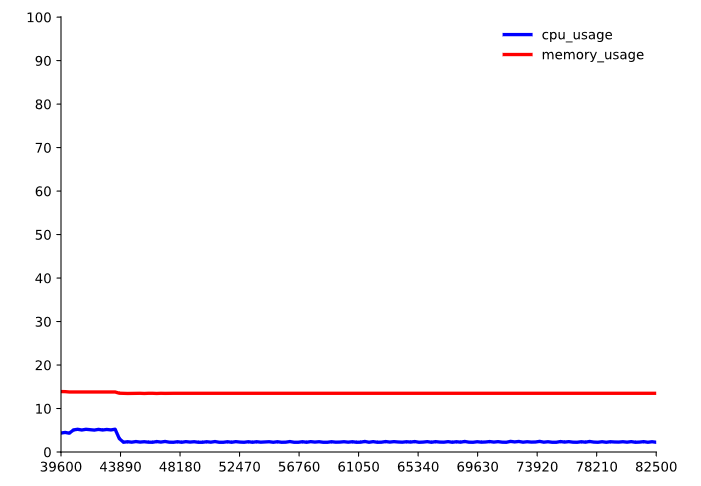


Figure 5 E类节点cpu\_usage/memory\_usage随时间的变化曲线

F类节点：cpu\_usage/memory\_usage在70000s左右时出现大概10000s的尖峰。例232号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

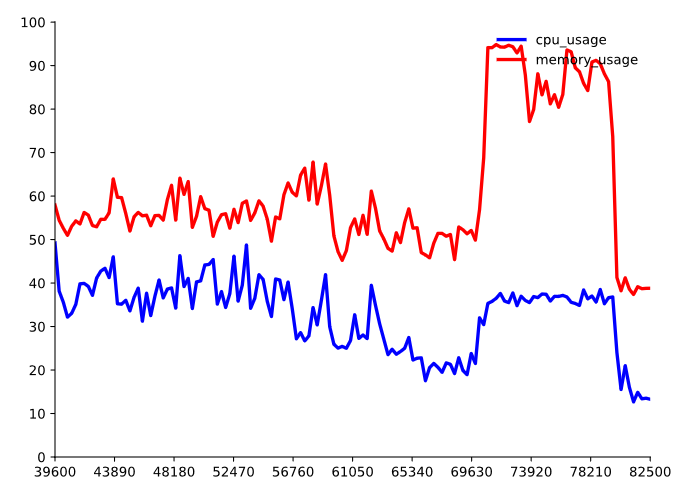


Figure 6 F类节点cpu\_usage/memory\_usage随时间的变化曲线

G类节点：cpu\_usage/memory\_usage在70000s左右时出现大概4000s的尖峰。例341号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

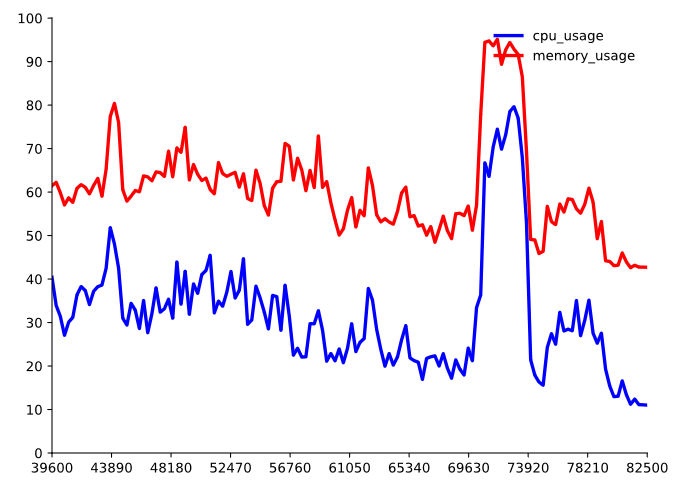


Figure 7 G类节点cpu\_usage/memory\_usage随时间的变化曲线

H类节点：cpu\_usage/memory\_usage在70000s左右时出现短暂的尖峰。例426号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

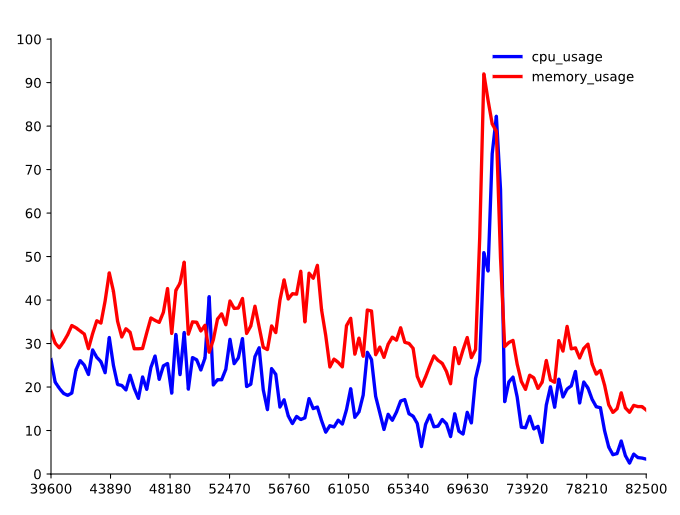


Figure 8 H类节点cpu\_usage/memory\_usage随时间的变化曲线

I类节点：cpu\_usage/memory\_usage在前10000s波动较小。例618号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

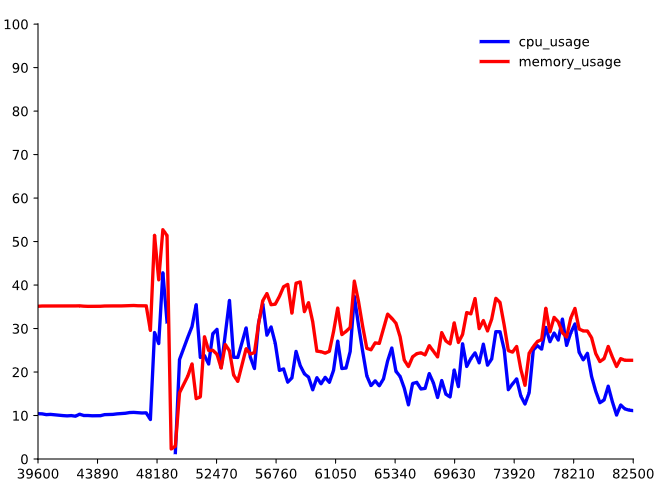


Figure 9 I类节点cpu\_usage/memory\_usage随时间的变化曲线

J类节点：cpu\_usage/memory\_usage在52000s左右出现5000s的水平。例683号节点，其cpu\_usage/memory\_usage随时间变化曲线如下所示。

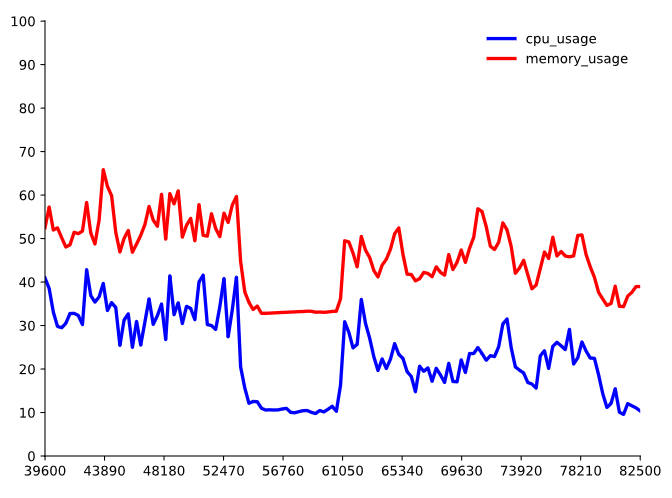


Figure 10 J类节点cpu\_usage/memory\_usage随时间的变化曲线

1. 不同类型节点的个数统计

我们对1313个节点各自属于那类节点进行了统计分析，各类节点的个数如表1所示，其占总结点个数的百分比如图11（蓝色柱形图）所示。我们可以看出A类节点数量为1111个，所占比例高达85%，C类节点个数为154，所占比例大概为10%，而其余类型节点只占极少数。

Table 1 不同类型节点个数

|  |  |
| --- | --- |
| 节点类型 | 节点数量 |
| A | 1111 |
| B | 2 |
| C | 154 |
| D | 3 |
| E | 17 |
| F | 4 |
| G | 10 |
| H | 7 |
| I | 1 |
| J | 4 |

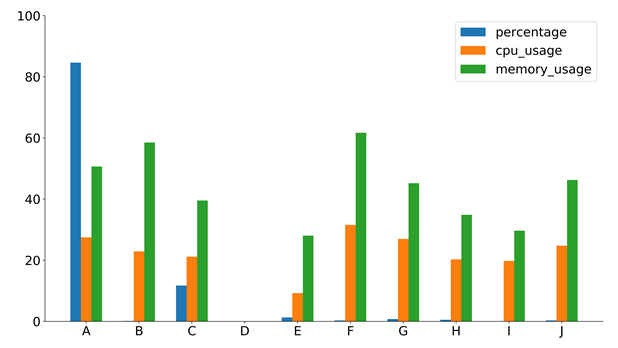


Figure 11 不同类型节点数量百分比，CPU平均利用率，memory平均利用率

从上图可以看出B类和F类节点的memory利用率较高，而E类节点的cpu利用率较低，我们可以根据这些特点对不同类节点进行任务调度。

1. 不同类型节点cpu\_usage/memory\_usage分析

为了分析各类节点cpu\_usage和memory\_usage的特点，我们计算了各类节点cpu\_usage和memory\_usage的平均值,中位值和方差，具体数字如表2所示，根据表2，我们画出了各类节点cpu和memory平均利用率的柱形图，如图11（橙色和绿色柱形图）以供我们进行分析。

Table 2 不同类型节点cpu\_usage和memory\_usage

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 节点类型 | cpu\_usage\_mean | mem\_usage  \_mean | cpu\_usage  \_median | mem\_usage  \_median | cpu\_usage  \_std | mem\_usage  \_std |
| A | 27.4 | 50.6 | 27.2 | 52.6 | 9.8 | 12.4 |
| B | 22.8 | 58.6 | 22.1 | 58.6 | 4.6 | 9.6 |
| C | 21.1 | 39.5 | 15.1 | 35.7 | 13.3 | 11.6 |
| D | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 9.26 | 28.0 | 9.7 | 30.5 | 6.6 | 12.8 |
| F | 31.5 | 61.6 | 33.2 | 57.0 | 9.5 | 15.9 |
| G | 26.9 | 45.1 | 24.0 | 44.4 | 14.9 | 16.4 |
| H | 20.2 | 34.8 | 19.1 | 31.3 | 11.5 | 14.5 |
| I | 19.7 | 29.6 | 19.0 | 29.5 | 7.6 | 7.4 |
| J | 24.7 | 46.2 | 24.42000008 | 46.3 | 9.1 | 8.4 |

## B． Job durations and distribution

这小节，我们从多个角度对Job进行了分析。下面讲一一描述我们的分析结果。

1. 每个Job的instance number 以及task number

根据我们已有的数据，我们知道对于不同的Job运行的instance和task的数目不同，为了得出每个Job上各自运行了多少instance和task,我们对已有数据进行了统计。统计结果如图12，图13所示。

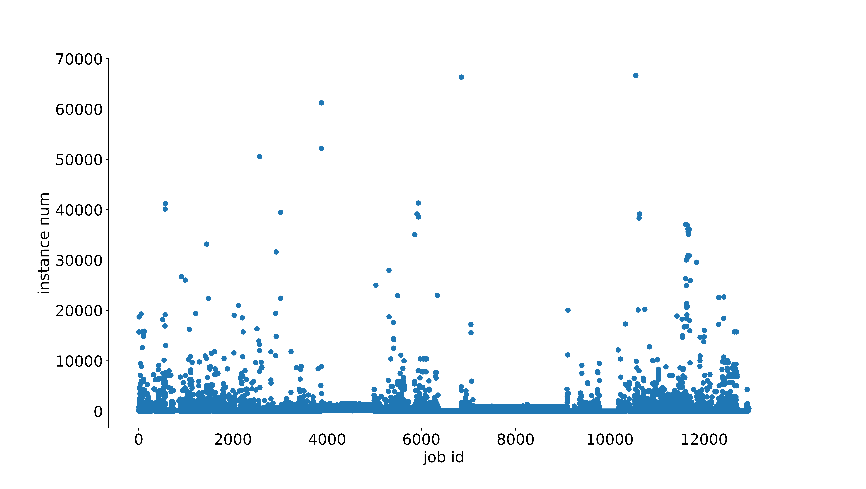


Figure 12 不同Job运行的instance number

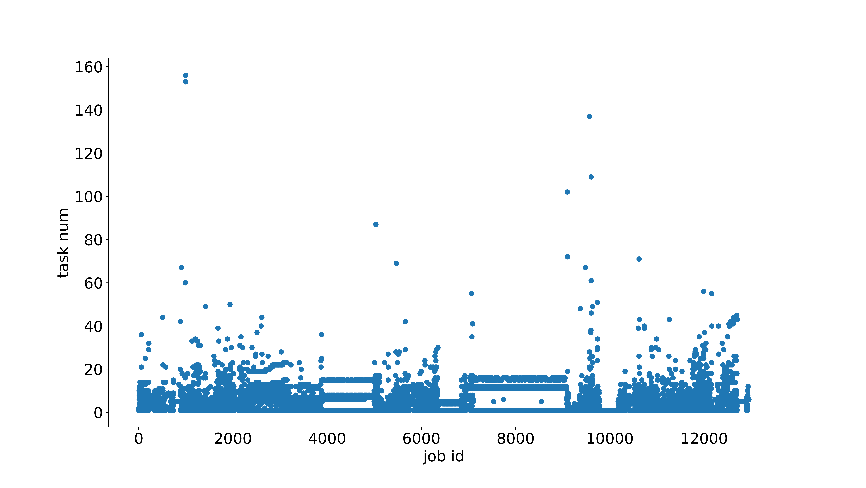


Figure 13 不同Job运行的task number

为进一步分析每个Job的instance number和task number的分布情况，我们做出了其分布的CDF图，如图14，图15所示。

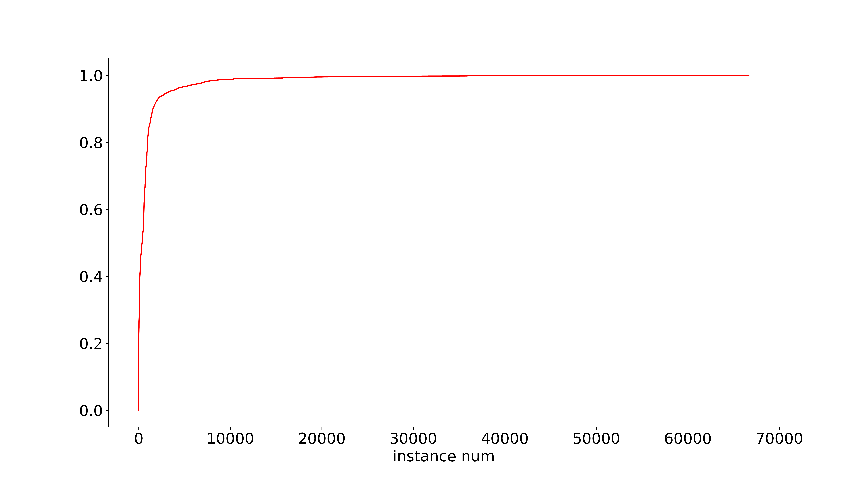


Figure 14 不同Job运行的instance number的CDF图

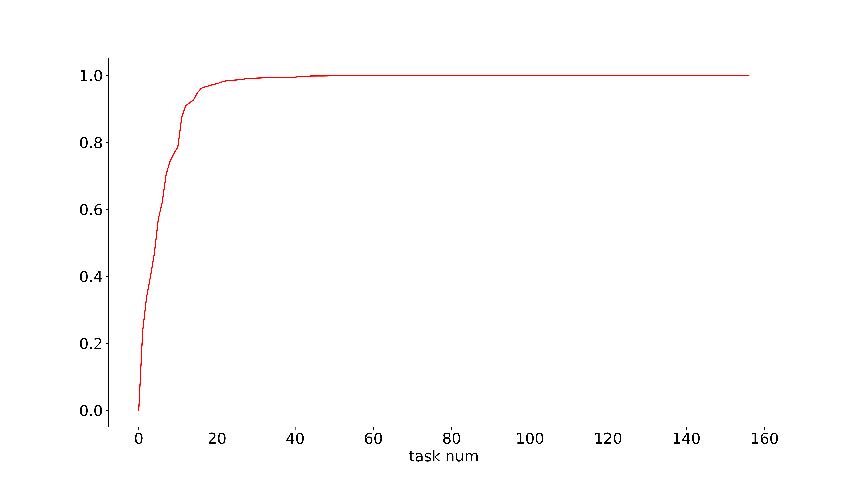


Figure 15 不同Job运行的task number的CDF图

从图12可以得出，对于大多数的Job，其运行的instance number集中在0-10000。图14也说明了这一情况，当instance\_number在0-10000时，概率曲线非常陡峭，当达到20000时，其概率已经趋近1。

对于Job运行的task number，同instance number类似，从图13可以得出，对于大多数的Job，其运行的task number集中在0-20。从图15可以看出，当task\_num在0-20时，概率曲线非常陡峭，当达到40时，其概率也趋紧1。

1. Job运行不同数目instance/task运行时间

因为每个Job上都运行了一定数目的instance和task，所以我们分析了每个Job运行不同数目instance/task时的运行时间，结果如图16,图17所示。从图16可以看出当Job运行的instance number在0-1000时，所需时间为一个增长的状态，而当instance number大于1000时，随着number的增加，运行时间明显下降。

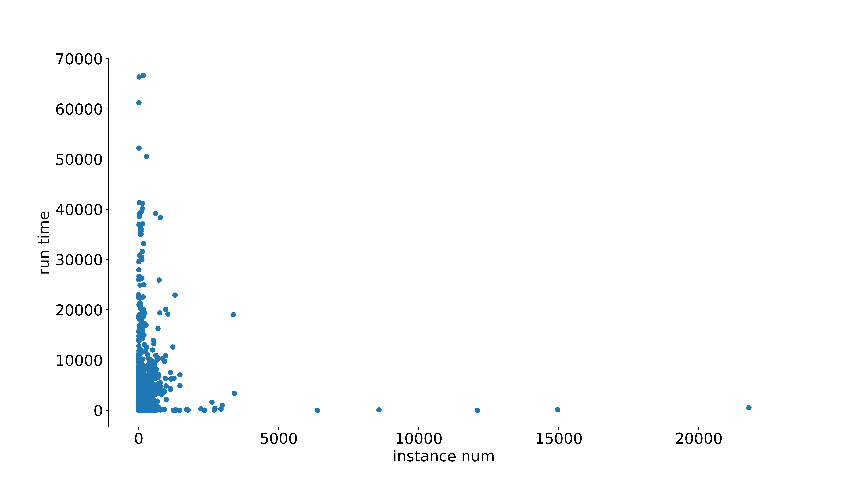


Figure 16 不同instance number运行时间

从图17我么可以看出，当Job的task number在0-5时，Job的运行时间比较长，而在其他number时，Job的运行时间都集中在0-2000s之间。

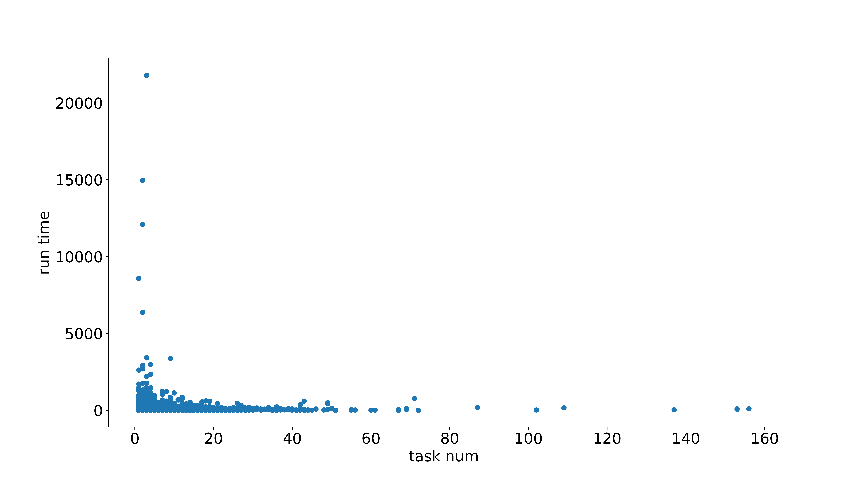


Figure 17 不同task number运行时间

1. 每个Job完成时间

最后我们分析了每个Job的完成时间，分析结果如图18所示。

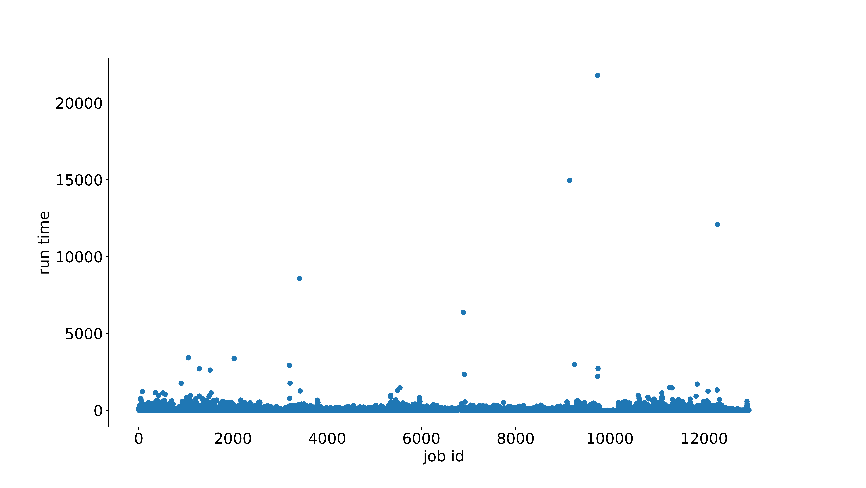


Figure 18 不同job的完成时间

## C. Correlations between Memory and CPU utilization

这节我们将针对节点的资源利用情况开展研究，从而在各类资源取得均衡。我们根据节点的cpu和memory的利用率，分析了二者之间的关系。

1. CPU\_usage与memory\_usage的相关系数

根据节点的cpu和memory的利用率，我们计算了cpu利用率与memory利用率的相关系数，来评估节点资源利用之间的关系。计算结果如图12所示。从图可以看出，所求的相关参数在[-1,1]之间，节点cpu和memory的利用率既有正相关也可有负相关，但大多数的节点分布在0.6-1之间，整体上看，CPU和memory的变化是正相关的。

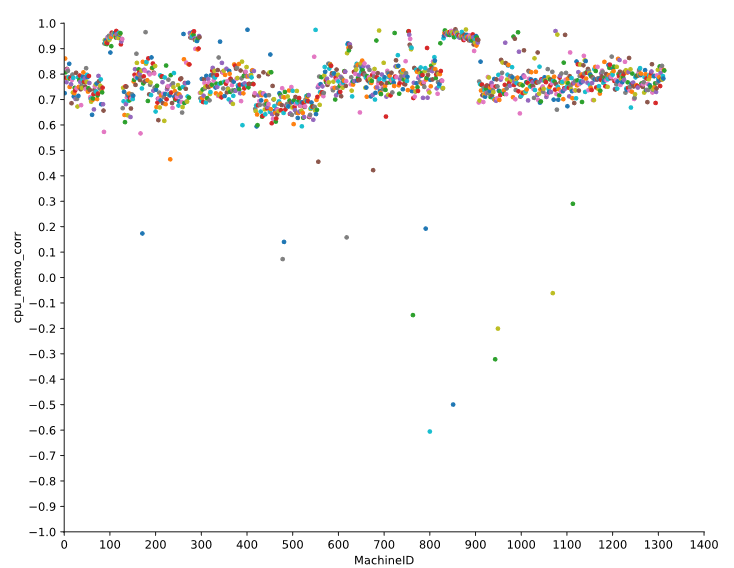


Figure 12 各节点cpu利用率与memory利用率的相关参数

1. B部分所介绍10类节点的cpu\_usage和memory\_usage的差值

为进一步分析各类型节点cpu\_usage和memory\_usage之间存在的关系，我们计算了二者之间的差值，并根据计算的结果，得出了各类型节点cpu\_usage和memory\_usage差值的特征。

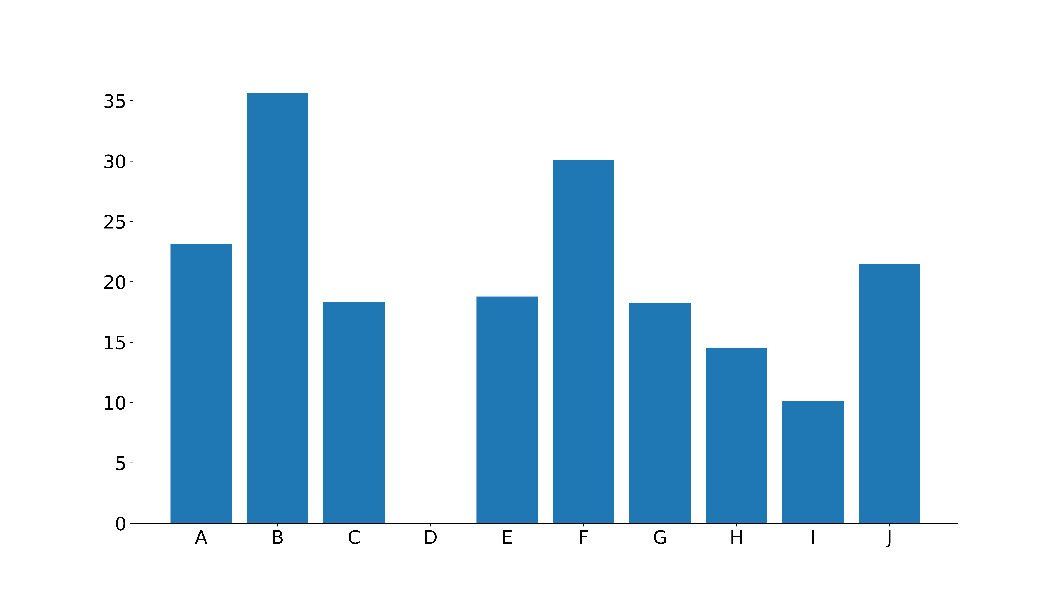


Figure13 各类节点cpu利用率和memory利用率的差值