exp_02

January 10, 2019

1 Data inspection

Below we can find the performance results of a mini benchmark `bench-stat` which were collected via `ci-hpc` framework.

The benchmark `bench-stat` was executed on a `charon` resource.

1.1 About `bench-stat` application

`bench-stat` application is a set of 3 benchmarks performing simple memory operations in a level 1, level 2 and level 3 cache.

These operations can be extremely fast, thus the experiments are *repeated* `N` times to obtain measurable duration of the benchmarks.

$$N = 1024 * 1024 * reps$$

where *reps* is a extra repetition coefficient which is altered in all of the commits.

As a baseline commit tagged as `reps-100` was selected, where $reps = 100$. The total number of repetition for this commit is $N = 1024 * 1024 * 100 = 104\,857\,600$

A commits with maximum and minumum number of *reps* are tagged as `reps-125` and `reps-075` respecively.

1.2 Data structure

In table below we can see a *simplified* format of the data collected. The most of the fields are self-explanitory however some of them require explanation:

- `tag` - a `git` tag of a commit making the results more human-readible
- `timepoint` - numerical value of a `tag` for further purposes
- `no` - i-th repetition

```
In [1]: %matplotlib inline
        from cihpc.exp import exp_02_init as env
        env = env.reload(env)
        np, sc, pd, plt, sea = env.np, env.sc, env.pd, env.plt, env.sea
        df = env.fetch_data()
        df.head()
```

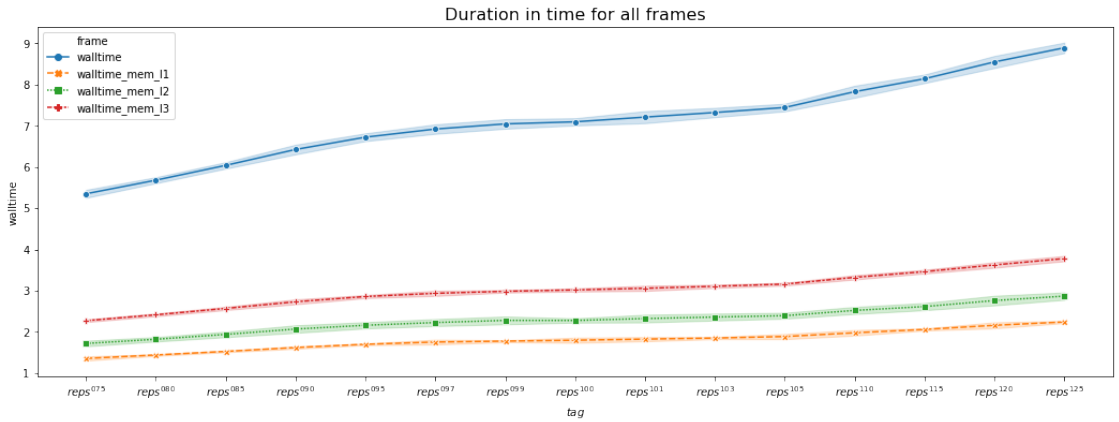
```
Out[1]:      commit  walltime  time       tag         $tag$  no  walltime_mem_l1  \
        0  75d38b25  5.420462    75  reps-075  $reps^{075}$   0         1.358015
        1  75d38b25  5.428641    75  reps-075  $reps^{075}$   1         1.373095
        2  75d38b25  5.359639    75  reps-075  $reps^{075}$   2         1.343519
        3  75d38b25  5.296926    75  reps-075  $reps^{075}$   3         1.343125
        4  75d38b25  5.333107    75  reps-075  $reps^{075}$   4         1.328733

           walltime_mem_l2  walltime_mem_l3
        0         1.808600         2.253832
        1         1.766966         2.288569
        2         1.784171         2.231938
        3         1.682919         2.270869
        4         1.763296         2.241055
```

1.3 Impact of individual commits on a duration

Chart below illustrates relation between `walltime [sec]` and commits, marked as

$$reps^{075}, reps^{080}, \ldots, reps^{095}, reps^{097}, reps^{099}, reps^{100}, reps^{101}, reps^{103}, reps^{105}, reps^{110}, \ldots, reps^{125}$$

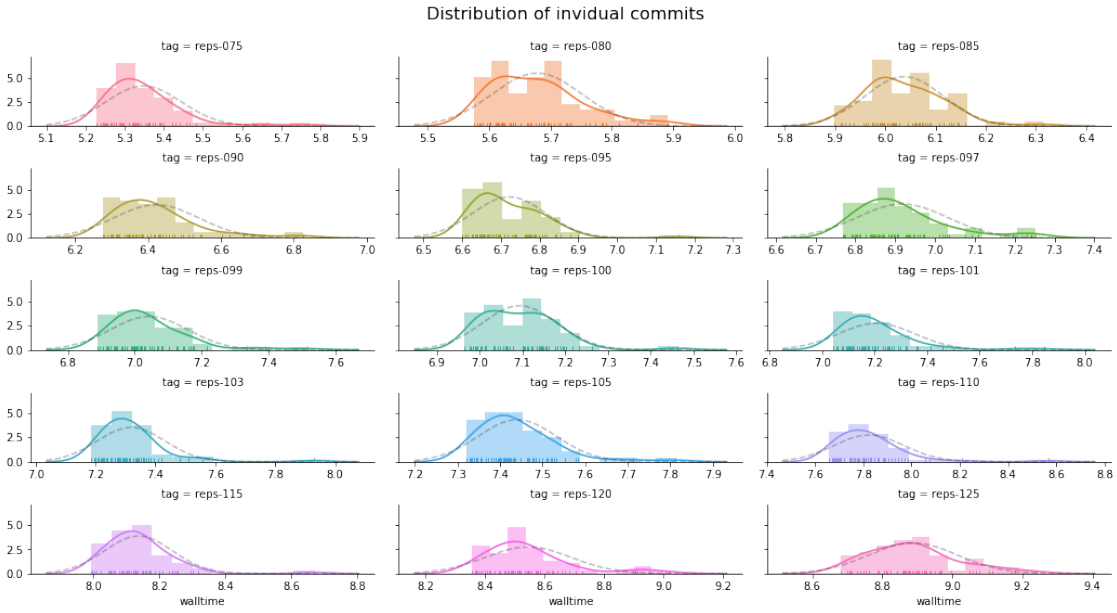
```
In [2]: df2 = env.unwrap(df, ['walltime', 'walltime_mem_l1', 'walltime_mem_l2', 'walltime_mem_
        plt.figure(figsize=(18, 6))
        sea.lineplot(data=df2, x='$tag$', y='walltime', style='frame', hue='frame', markers=Tru
        plt.title('Duration in time for all frames', size=16);
```



1.4 Data distribution for individual commits

Charts below show histogram for each of the 15 commits along with `normal` fit (gray dashed line)

```
In [3]: g = sea.FacetGrid(df, col='tag', col_wrap=3, aspect=3, height=1.6, sharex=False, hue='t
        g.map(sea.distplot, 'walltime', bins=11, rug=True, fit=sc.stats.norm, fit_kws=env.alpha
        g.fig.suptitle("Distribution of invidual commits", size=16)
        g.fig.subplots_adjust(top=.9)
```


Distribution of invidual commits
tag = reps-075
tag = reps-080
tag = reps-085
tag = reps-090
tag = reps-095
tag = reps-097
tag = reps-099
tag = reps-100
tag = reps-101
tag = reps-103
tag = reps-105
tag = reps-110
tag = reps-115
tag = reps-120
tag = reps-125
walltime
walltime
walltime