Hatchet: Pruning the Overgrowth in Parallel Profiles

By Abhinav Bhatele, Stephanie Brink, and Todd Gamblin

XIANGYU MAO

2/18/2021

Limitations of Profiling Tools

- Unique format
- Limited display (GUI) options
- Limited supported analysis
  - Number of call graphs
  - Details in subsections
- Limited Programmability
- Dependency between measurement and analysis tools

Pandas: Basic Data Structures

An open-source Python library for data analysis

- **Series**
  - 1D
  - Homogeneously-typed
  - Indexed
  - Similar to Hashtable

- **DataFrame**
  - 2D
  - Heterogeneously-typed
  - Each column as a series
  - Spreadsheet / SQL functionalities

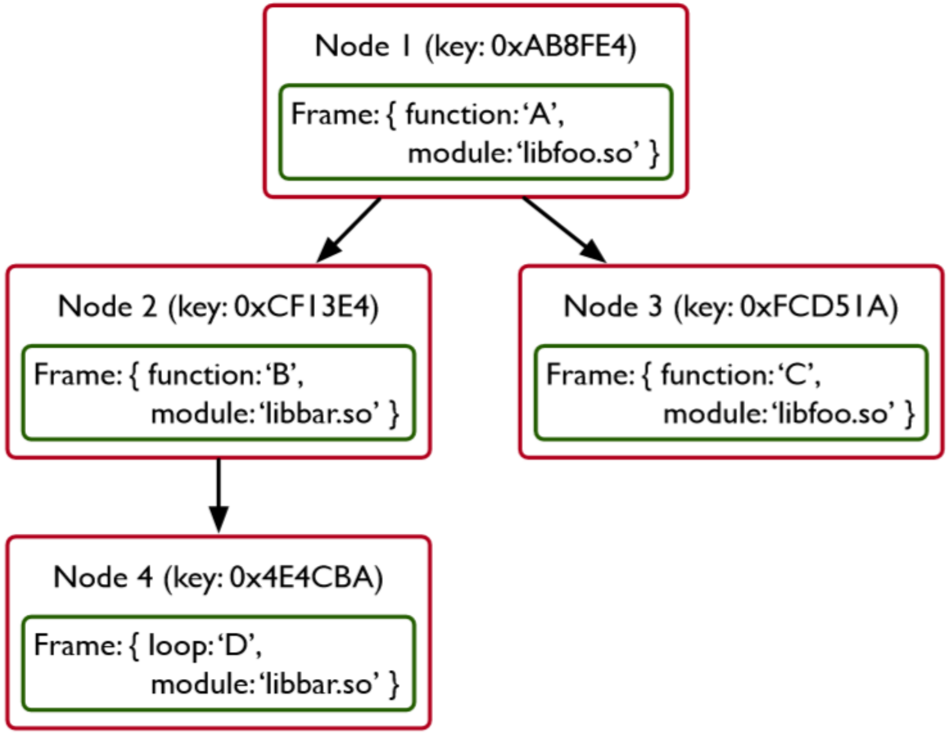
- **MultiIndex**
  - Multiple columns in DataFrame
  - Multidimensional data manipulation

Hatchet: Structured Index

- Based on Pandas
- Provides non-linear data structures

  (Pandas can only handle linear data)
- Tree/Graph node → DataFrame index

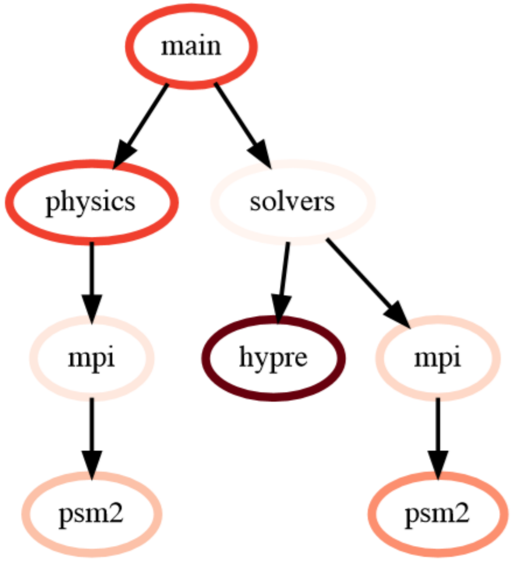
- **Frame**: code construct
- Node: connections

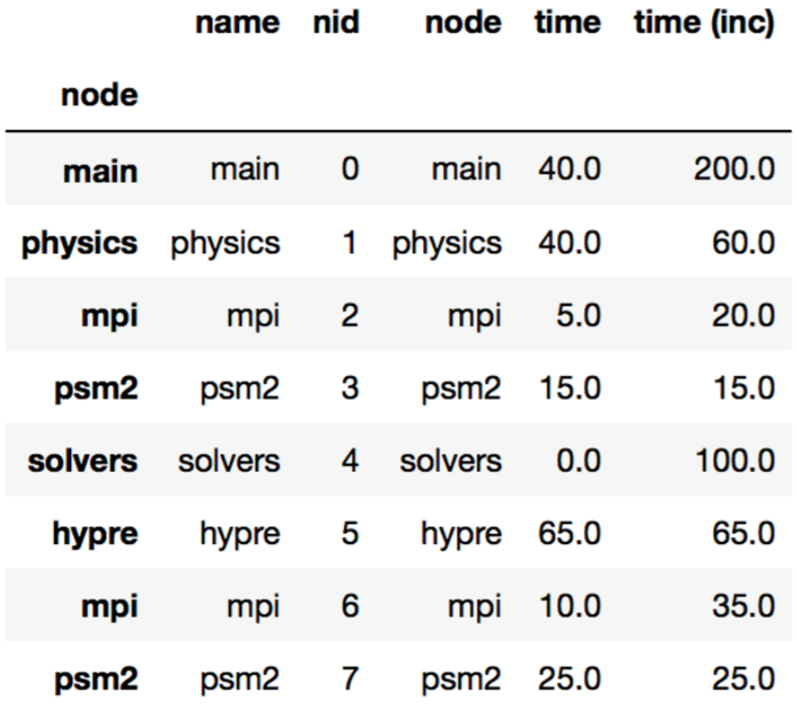

Node 1 (key: 0xAB8FE4)
Frame: { function: 'A', module: 'libfoo.so' }
Node 2 (key: 0xCF13E4)
Frame: { function: 'B', module: 'libbar.so' }
Node 3 (key: 0xFCD51A)
Frame: { function: 'C', module: 'libfoo.so' }
Node 4 (key: 0x4E4CBA)
Frame: { loop: 'D', module: 'libbar.so' }

Key comparators can be defined for frame and node separately

Hatchet: GraphFrame

- Structured index & DataFrame Combined
- Nodes used as index column

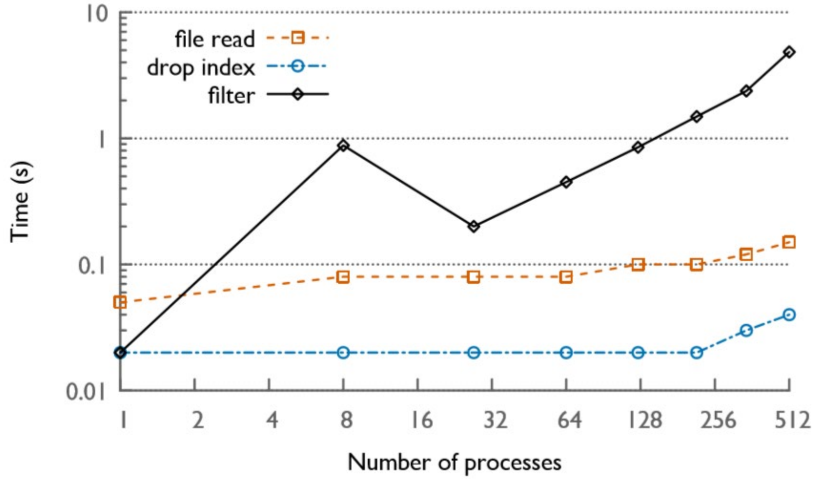




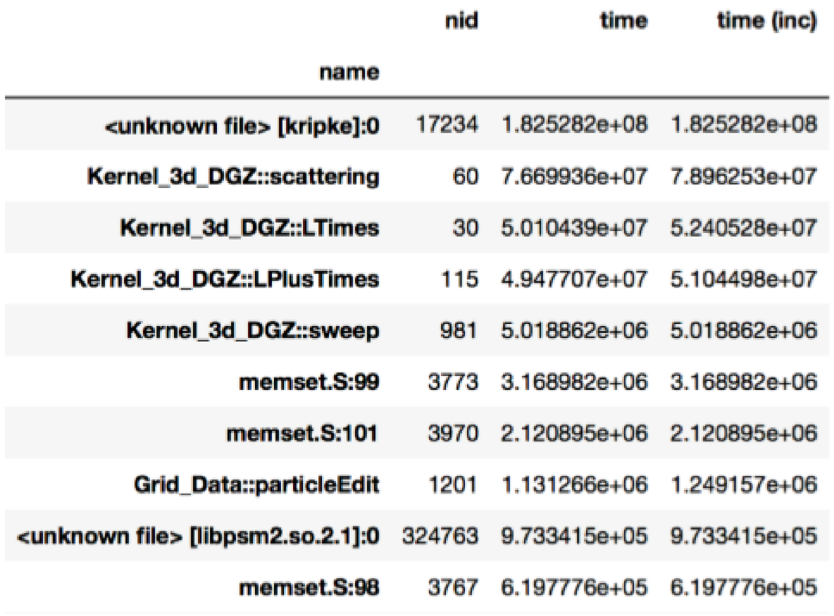
	name	nid	node	time	time (inc)
node					
main	main	0	main	40.0	200.0
physics	physics	1	physics	40.0	60.0
mpi	mpi	2	mpi	5.0	20.0
psm2	psm2	3	psm2	15.0	15.0
solvers	solvers	4	solvers	0.0	100.0
hypre	hypre	5	hypre	65.0	65.0
mpi	mpi	6	mpi	10.0	35.0
psm2	psm2	7	psm2	25.0	25.0

Operations

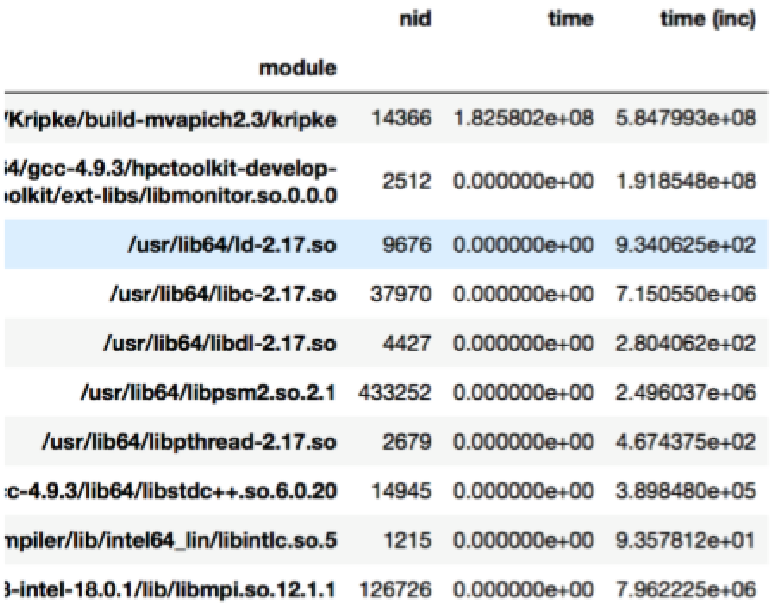
- **DataFrame**
  - filter
  - drop_index_levels
  - update_inclusive_columns
- **Graph**
  - squash
  - equal
  - union
- **GraphFrame**
  - copy
  - unify
  - add
  - subtract



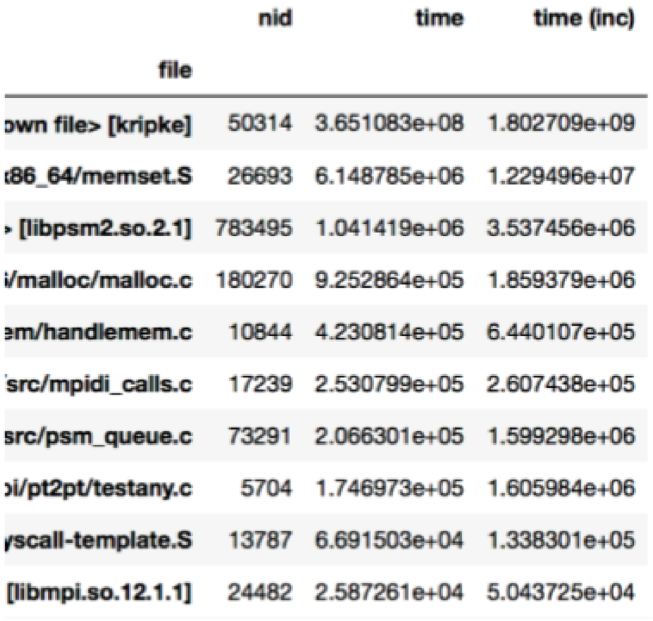
Analysis: Flat Profiles



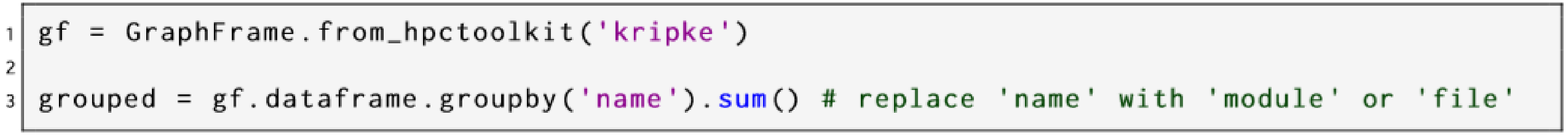
name	nid	time	time (inc)
<unknown file> [kripke]:0	17234	1.825282e+08	1.825282e+08
Kernel_3d_DGZ::scattering	60	7.669936e+07	7.896253e+07
Kernel_3d_DGZ::LTimes	30	5.010439e+07	5.240528e+07
Kernel_3d_DGZ::LPlusTimes	115	4.947707e+07	5.104498e+07
Kernel_3d_DGZ::sweep	981	5.018862e+06	5.018862e+06
memset.S:99	3773	3.168982e+06	3.168982e+06
memset.S:101	3970	2.120895e+06	2.120895e+06
Grid_Data::particleEdit	1201	1.131266e+06	1.249157e+06
<unknown file> [libpsm2.so.2.1]:0	324763	9.733415e+05	9.733415e+05
memset.S:98	3767	6.197776e+05	6.197776e+05



module	nid	time	time (inc)
/Kripke/build-mvapich2.3/kripke	14366	1.825802e+08	5.847993e+08
4/gcc-4.9.3/hpctoolkit-develop- olkit/ext-libs/libmonitor.so.0.0.0	2512	0.000000e+00	1.918548e+08
/usr/lib64/ld-2.17.so	9676	0.000000e+00	9.340625e+02
/usr/lib64/libc-2.17.so	37970	0.000000e+00	7.150550e+06
/usr/lib64/libdl-2.17.so	4427	0.000000e+00	2.804062e+02
/usr/lib64/libpsm2.so.2.1	433252	0.000000e+00	2.496037e+06
/usr/lib64/libpthread-2.17.so	2679	0.000000e+00	4.674375e+02
c-4.9.3/lib64/libstdc++.so.6.0.20	14945	0.000000e+00	3.898480e+05
npiler/lib/intel64_lin/libintlc.so.5	1215	0.000000e+00	9.357812e+01
3-intel-18.0.1/lib/libmpi.so.12.1.1	126726	0.000000e+00	7.962225e+06

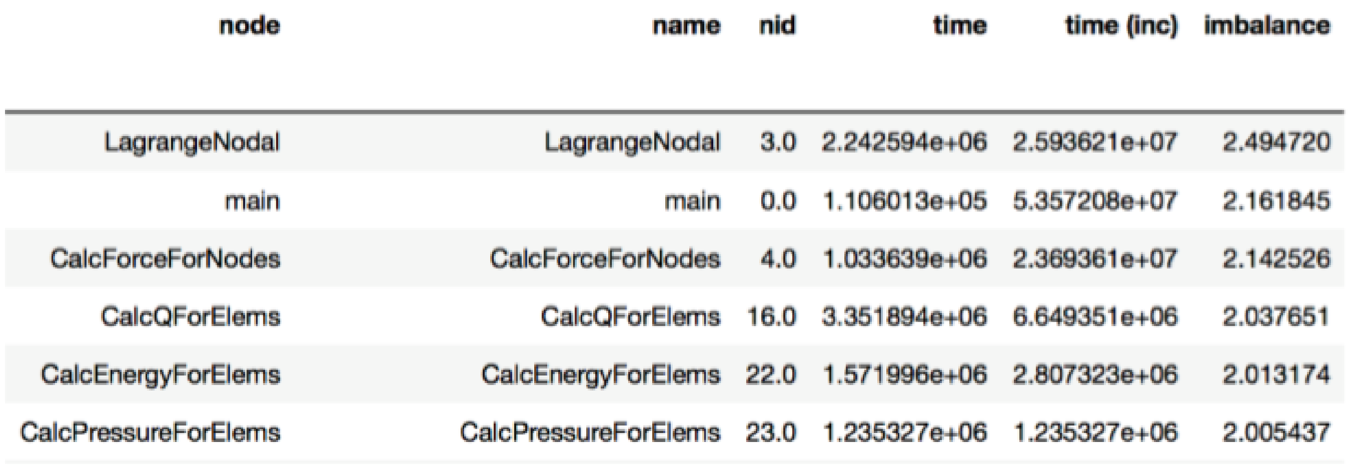


file	nid	time	time (inc)
own file> [kripke]	50314	3.651083e+08	1.802709e+09
x86_64/memset.S	26693	6.148785e+06	1.229496e+07
> [libpsm2.so.2.1]	783495	1.041419e+06	3.537456e+06
/malloc/malloc.c	180270	9.252864e+05	1.859379e+06
em/handlemem.c	10844	4.230814e+05	6.440107e+05
src/mpidi_calls.c	17239	2.530799e+05	2.607438e+05
src/psm_queue.c	73291	2.066301e+05	1.599298e+06
i/pt2pt/testany.c	5704	1.746973e+05	1.605984e+06
yscall-template.S	13787	6.691503e+04	1.338301e+05
[libmpi.so.12.1.1]	24482	2.587261e+04	5.043725e+04

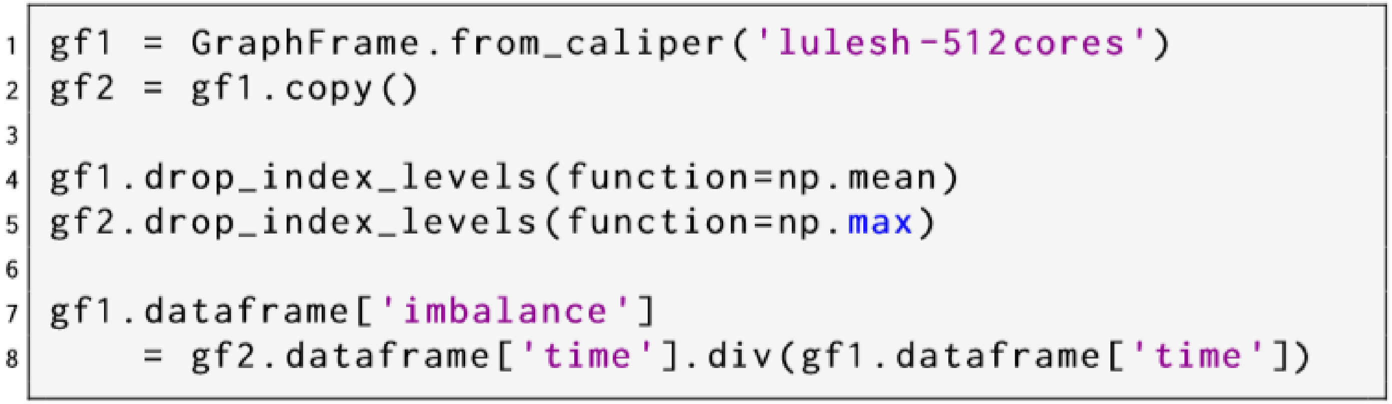


```python
gf = GraphFrame.from_hpctoolkit('kripke')

grouped = gf.dataframe.groupby('name').sum() # replace 'name' with 'module' or 'file'
```

Analysis: Load Imbalance

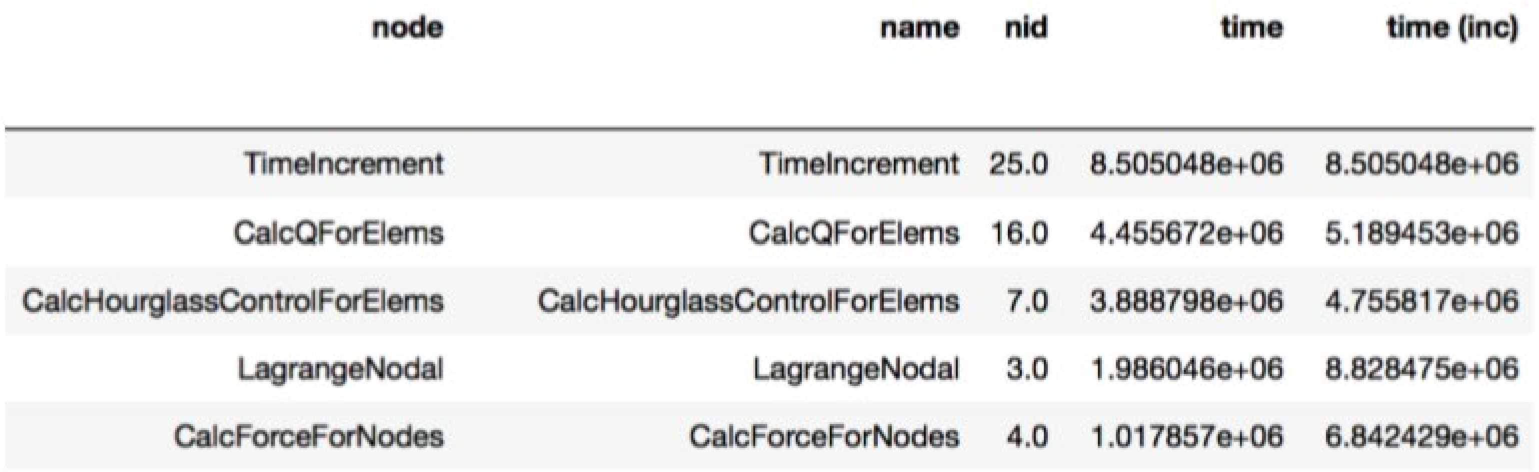


node	name	nid	time	time (inc)	imbalance
LagrangeNodal	LagrangeNodal	3.0	2.242594e+06	2.593621e+07	2.494720
main	main	0.0	1.106013e+05	5.357208e+07	2.161845
CalcForceForNodes	CalcForceForNodes	4.0	1.033639e+06	2.369361e+07	2.142526
CalcQForElems	CalcQForElems	16.0	3.351894e+06	6.649351e+06	2.037651
CalcEnergyForElems	CalcEnergyForElems	22.0	1.571996e+06	2.807323e+06	2.013174
CalcPressureForElems	CalcPressureForElems	23.0	1.235327e+06	1.235327e+06	2.005437

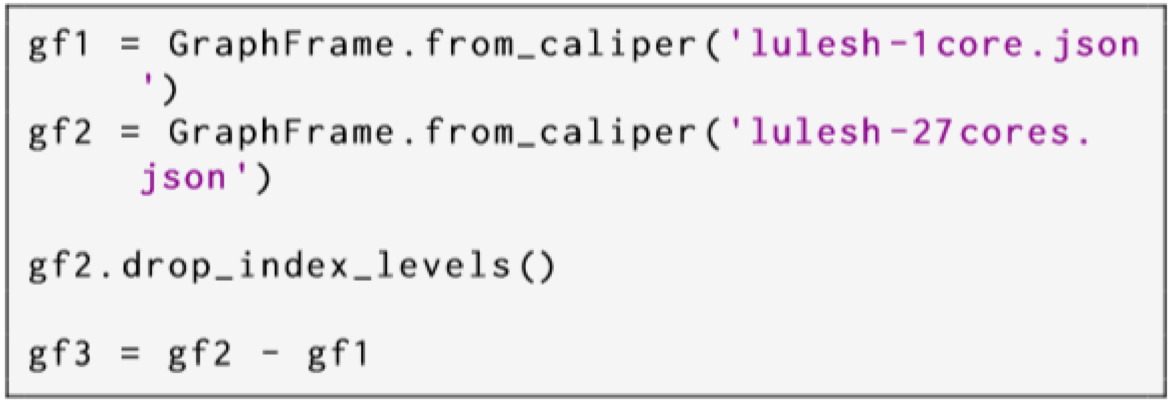


```
gf1 = GraphFrame.from_caliper('lulesh-512cores')
gf2 = gf1.copy()

gf1.drop_index_levels(function=np.mean)
gf2.drop_index_levels(function=np.max)

gf1.dataframe['imbalance']
    = gf2.dataframe['time'].div(gf1.dataframe['time'])
```

Analysis: Execution Profile Comparison

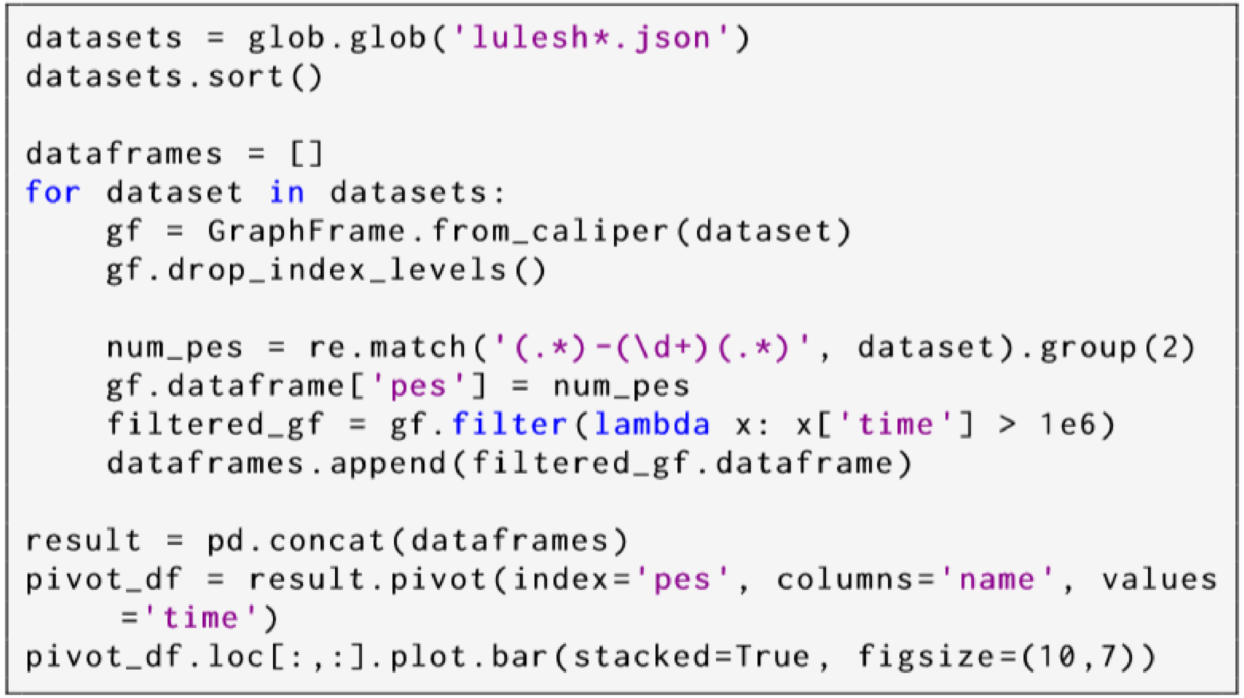


node	name	nid	time	time (inc)
TimeIncrement	TimeIncrement	25.0	8.505048e+06	8.505048e+06
CalcQForElems	CalcQForElems	16.0	4.455672e+06	5.189453e+06
CalcHourglassControlForElems	CalcHourglassControlForElems	7.0	3.888798e+06	4.755817e+06
LagrangeNodal	LagrangeNodal	3.0	1.986046e+06	8.828475e+06
CalcForceForNodes	CalcForceForNodes	4.0	1.017857e+06	6.842429e+06

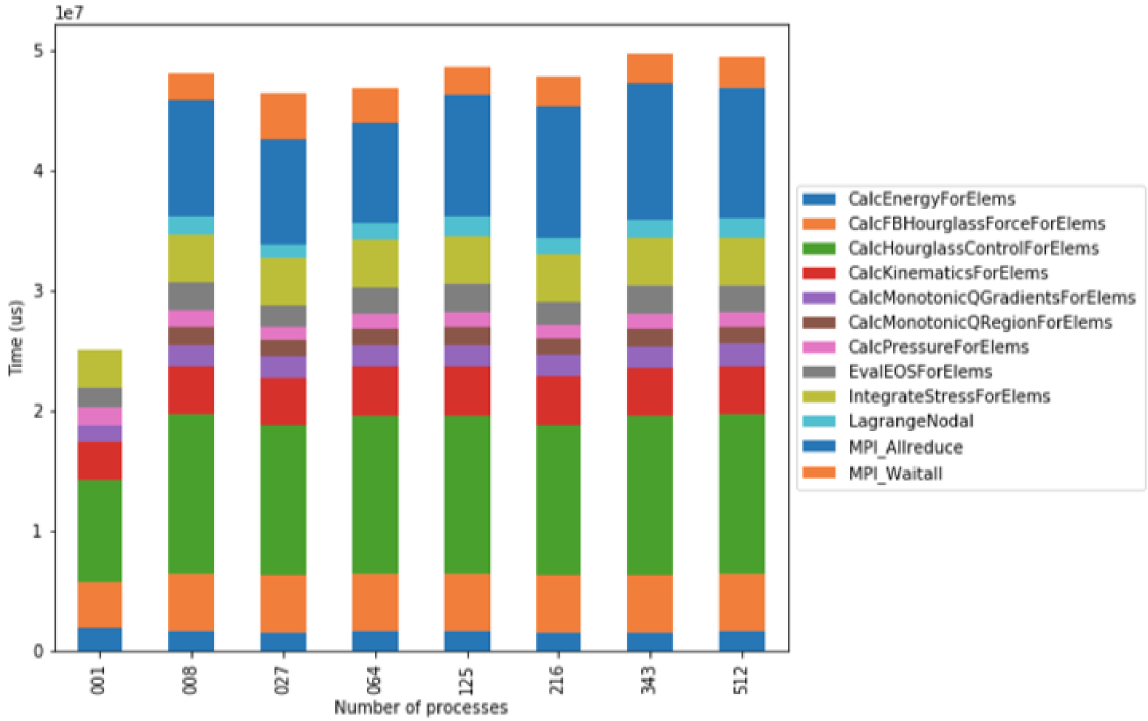


```
gf1 = GraphFrame.from_caliper('lulesh-1core.json')
gf2 = GraphFrame.from_caliper('lulesh-27cores.json')

gf2.drop_index_levels()

gf3 = gf2 - gf1
```

Analysis: Combination & Visualization



```
datasets = glob.glob('lulesh*.json')
datasets.sort()

dataframes = []
for dataset in datasets:
    gf = GraphFrame.from_caliper(dataset)
    gf.drop_index_levels()

    num_pes = re.match('(.*)-(\d+)(.*)', dataset).group(2)
    gf.dataframe['pes'] = num_pes
    filtered_gf = gf.filter(lambda x: x['time'] > 1e6)
    dataframes.append(filtered_gf.dataframe)

result = pd.concat(dataframes)
pivot_df = result.pivot(index='pes', columns='name', values
     ='time')
pivot_df.loc[:,:].plot.bar(stacked=True, figsize=(10,7))
```



Hatchet Conclusion

- Performance analysis tool
- Based on Pandas
- Added support for tree/graph structures
- Highly programmable by user
- Functionalities for various analysis
- Low overhead for operations