VisPerf:

visualize and compare performance of stream processing applications

Claudio Scheer[1] Isabel Harb Manssour[1] Dalvan Griebler[1]

[1]Pontifical Catholic University of Rio Grande do Sul - PUCRS
Brazil

Data Visualization, 2021/1

Table of Contents

1. Problem
2. Perf
3. VisPerf
4. Conclusions

Table of Contents

1. Problem
2. Perf
3. VisPerf
4. Conclusions

Stream Processing

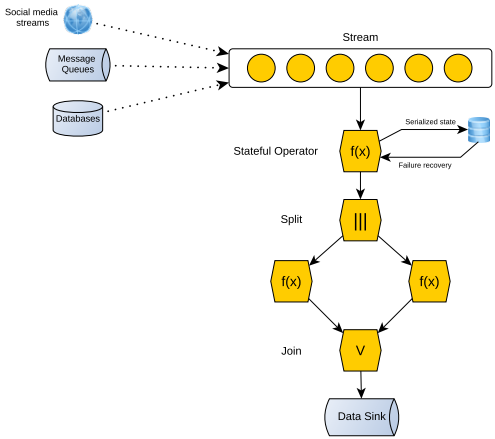

Figure: Stream processing operators.

Thread Placement

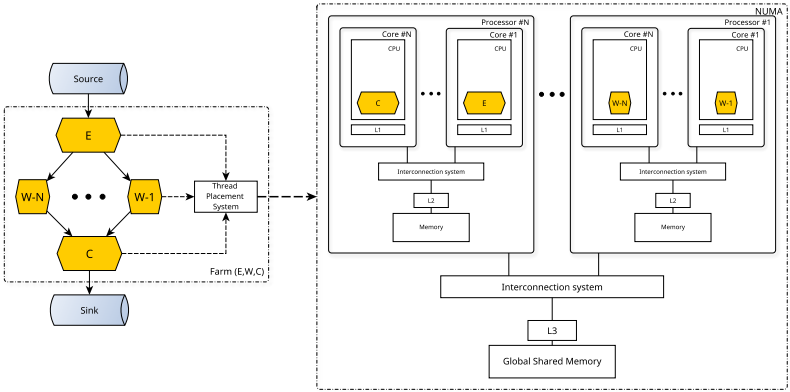

Figure: Stream processing placement 1.

Thread Placement

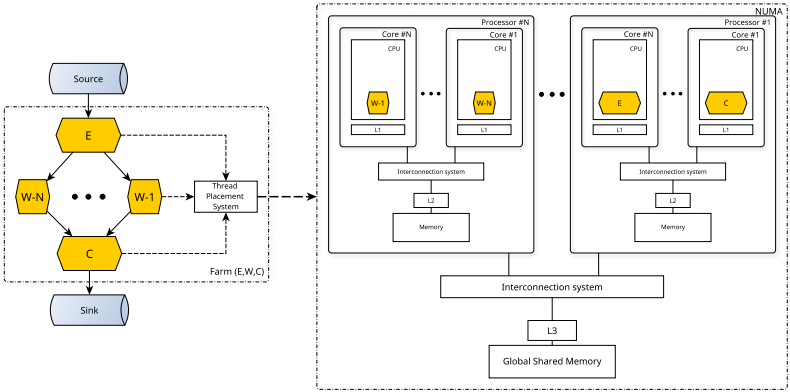

Figure: Stream processing placement 2.

Table of Contents

1 Problem

2 Perf

3 VisPerf

4 Conclusions

Perf

Linux tool to profile CPU performance counters.

`perf list`

- cache-misses
- instructions
- cpu-cycles
- L1-dcache-load-misses
- LLC-load-misses
- branch-misses
- bus-cycles
- ...

Compiling application

- `-g`
- `-fno-omit-frame-pointer`

`perf record -- sleep 10`

- `--sample-cpu`
- `--freq 997`
- `--events xxx,yyy`
- `--call-graph dwarf`

Perf Output File

```
ff-farm 32905 [002] 13134.421023:          19          cache-misses:
        55bb09b38ce5 ff::ff_node::thWorker::svc_init+0x1b (.../ff-farm)
        55bb09b3667d ff::ff_thread::thread_routine+0x3b (.../ff-farm)
        55bb09b341ae ff::proxy_thread_routine+0x23 (.../ff-farm)

ff-farm 32901 [011] 13134.421079:       36331       cache-references:
        7fc88cc40726 __memmove_avx_unaligned_erms+0xb6 (.../libc-2.31.so)
        7fc88ceb58b4 __pthread_attr_setaffinity_new+0x44 (inlined)
        55bb09b36486 ff::init_thread_affinity+0x162 (.../ff-farm)

...
```

Perf Output File

time	tid	event	counter	cpu	stack
13134.421023	32905	cache-misses	19	2	ff::ff_node::thWorker::svc_init| ff::ff_thread::thread_routine| ff::proxy_thread_routine
13134.421079	32901	cache-references	36331	11	__memmove_avx_unaligned_erms| __pthread_attr_setaffinity_new| ff::init_thread_affinity

...

Table of Contents

1 Problem

2 Perf

3 VisPerf

4 Conclusions

VisPerf

About

- Visualize captured PMUs;
- Compare different executions of the same application;
- See ‘regions of interest’;

Technologies

- Bash;
- Python;
- JavaScript;
- D3;
- React;

Demo

https://gmap.pucrs.br/claudioscheer/visperf

Source code

https://github.com/GMAP/visperf

Table of Contents

1 Problem

2 Perf

3 VisPerf

4 Conclusions

Conclusions

- It is only a prototype;
- Collect more performance metrics;
  - Power usage, for example;
- Include more evaluation metrics;
- Add more visualizations;
- Take feedbacks from other developers;

References

- https://perf.wiki.kernel.org/index.php/Main_Page
- https://www.brendangregg.com/perf.html
- https://man7.org/linux/man-pages/man1/perf.1.html