

# Detecting Memory-Bound Parallel Regions to Improve the Energy-Efficiency of Applications

Gabriel B. Moro and Lucas Mello Schnorr  
{gbmoro,schnorr}@inf.ufrgs.br



WSPPD 2017 – Porto Alegre, 4th September 2017

# Problem

- **Energy consumption** is an important issue in large data centers, battery-dependent devices, personal computers, and others.
- There is a **trade-off** between performance and energy consumption.

# Main Goal

- All programs have Memory-Bound regions, one of the ways to improve the energy consumption is to use processor frequencies adequate in that regions
  - To analyze the Lulesh application's behavior using a pre-run that provides the information needed to identify its Memory-Bound Regions



# Related Work

- Sequential Applications

- Spiliopoulos et al. [1] show a tool for sequential applications using phase detection, DVFS and correlation models.

- Parallel Applications

- Laurenzano et al. [2] define an automated approach that allows selecting the most appropriate processor frequency for a given program loop using metrics about application's behavior.

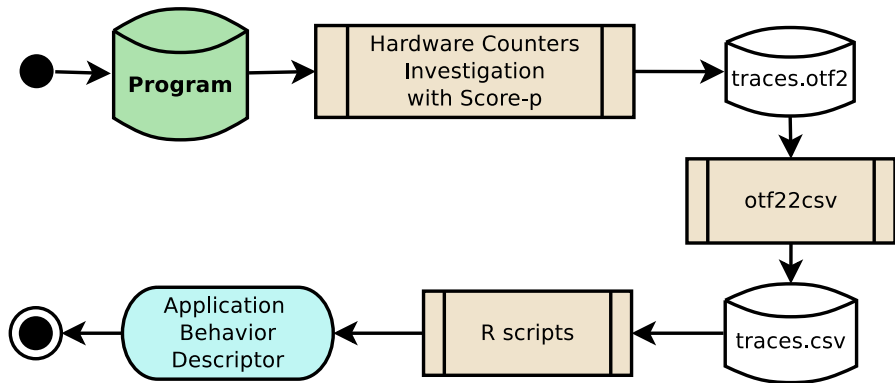
- Distributed Applications

- Freeh et al., [3] present an approach to distributed memory using adequate frequency for each node.

- OpenMP library

Millani and Schnorr [4] employs DoE (Design of Experiments) with Screening Designs. The approach consists in manual code instrumentation to signalize the parallel region in the source code to reducing the energy consumption.

# Methodology



# Experiment

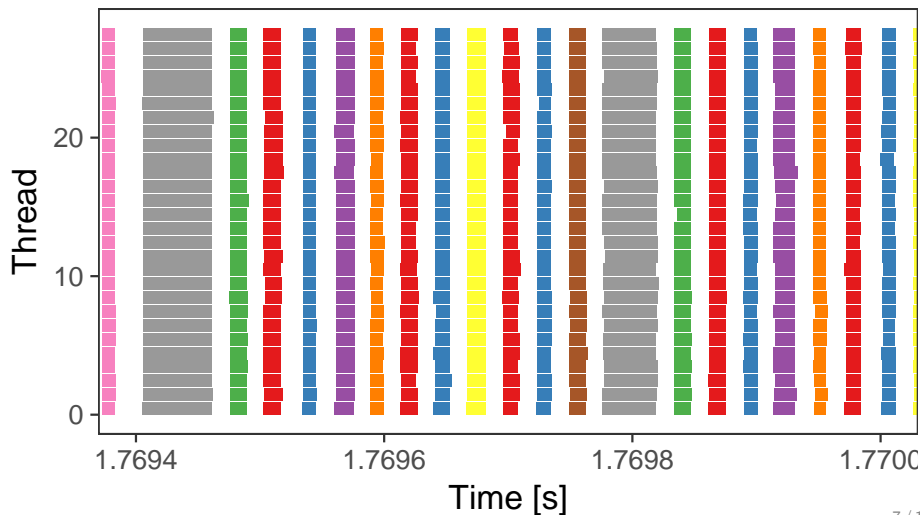
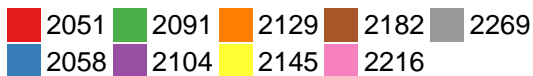
## Platform

- Hype2
  - 2 processors Intel(R) Xeon(R) CPU E5-2650 2.30GHz
    - 20 physical processors
    - Hyper Threading
  - Memory
    - L1 cache : 32K / L2: 256K / L3: 25600K
    - 126G of RAM Memory

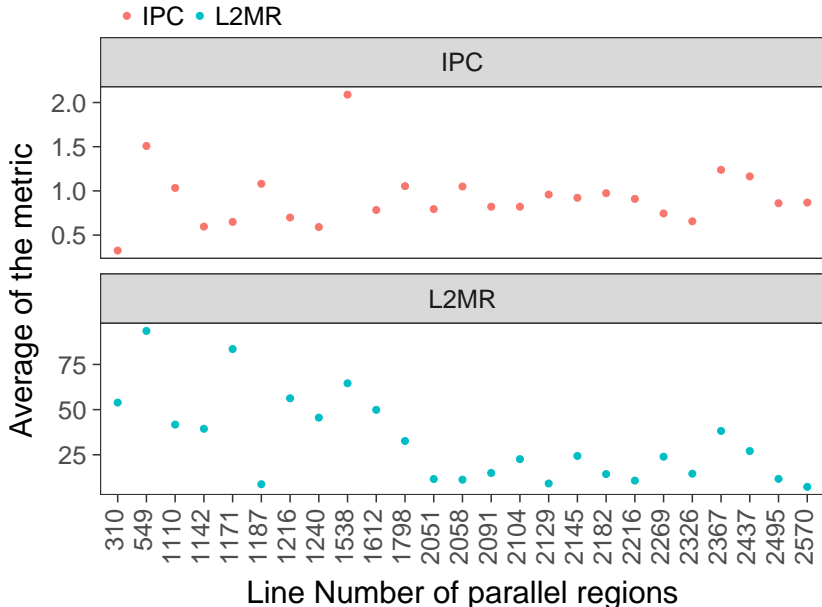
## Application

- Application: Lulesh
- 1 execution in **controlled environment**
  - Max processor frequency
  - Threads fixed
  - Turbo boost enabled

# Lulesh Behavior - Overview of Parallel Regions



# Lulesh - IPC and L2RM By Region



# Conclusion and Future Work

## Conclusion

- Trace each region, collecting HW counters
  - IPC: Instructions per Cycle
  - L2MR: L2 Miss rate
- [very] large traces

# Conclusion and Future Work

## Conclusion

- Trace each region, collecting HW counters
  - IPC: Instructions per Cycle
  - L2MR: L2 Miss rate
- [very] large traces

## Future work

- `libnina` (name yet to be defined)
  - Read a CSV file with region identifier and frequency
  - When entering that region, set the frequency
- Expectation
  - Energy reduction without performance loss

# Thanks for your attention

- Questions?