MAAT: EFFORTLESSLY SQUEEZING PERFORMANCE OUT OF HETEROGENEOUS SYSTEMS

Jose Luis Bosque, Borja Pérez, Esteban Stafford.

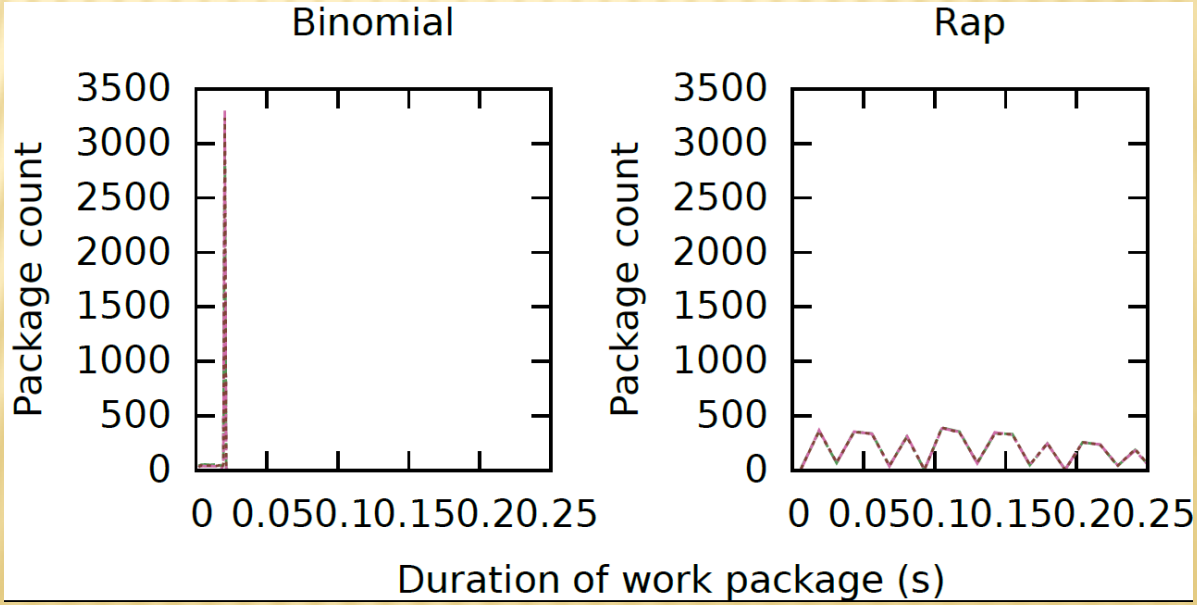
Universidad de Cantabria

B. Pérez, JL. Bosque, R. Beivide: Simplifying programming and load balancing of data parallel applications on heterogeneous systems. GPGPU@PPoPP 2016: 42-51

B. Pérez, E. Stafford, JL. Bosque, R. Beivide: Energy efficiency of load balancing for data-parallel applications in heterogeneous systems. The Journal of Supercomputing 73(1): 330-342 (2017)

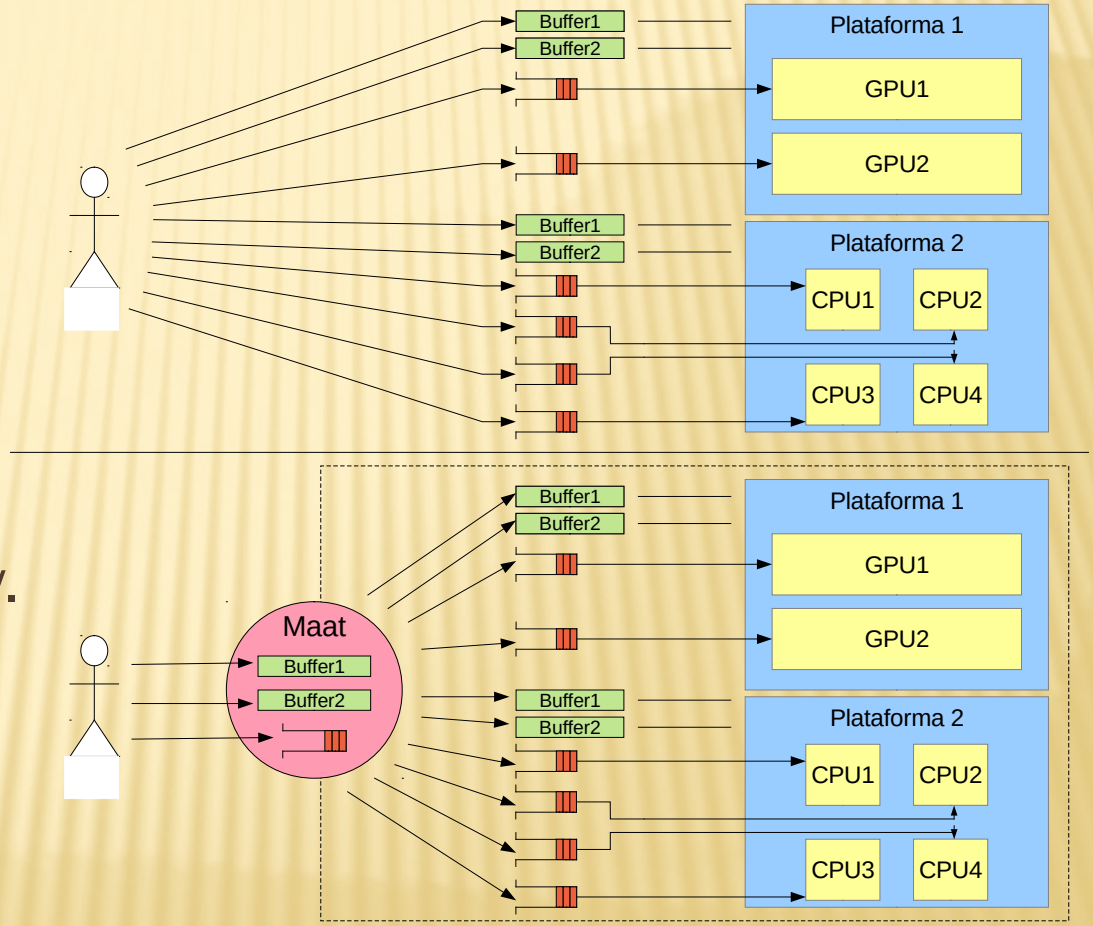
OBJECTIVES AND CHALLENGES

- **Performance portability**: to simplify the task of efficiently executing a single OpenCL kernel using all the devices in heterogeneous systems.
- **Challenges**:
  - Managing disjoint memory spaces through the concept of buffers.
  - To distribute data proportionally to the computational power of the individual devices.
  - Considering irregular applications: each unit of workload requires a different amount of computing time, not necessarily known a priori.



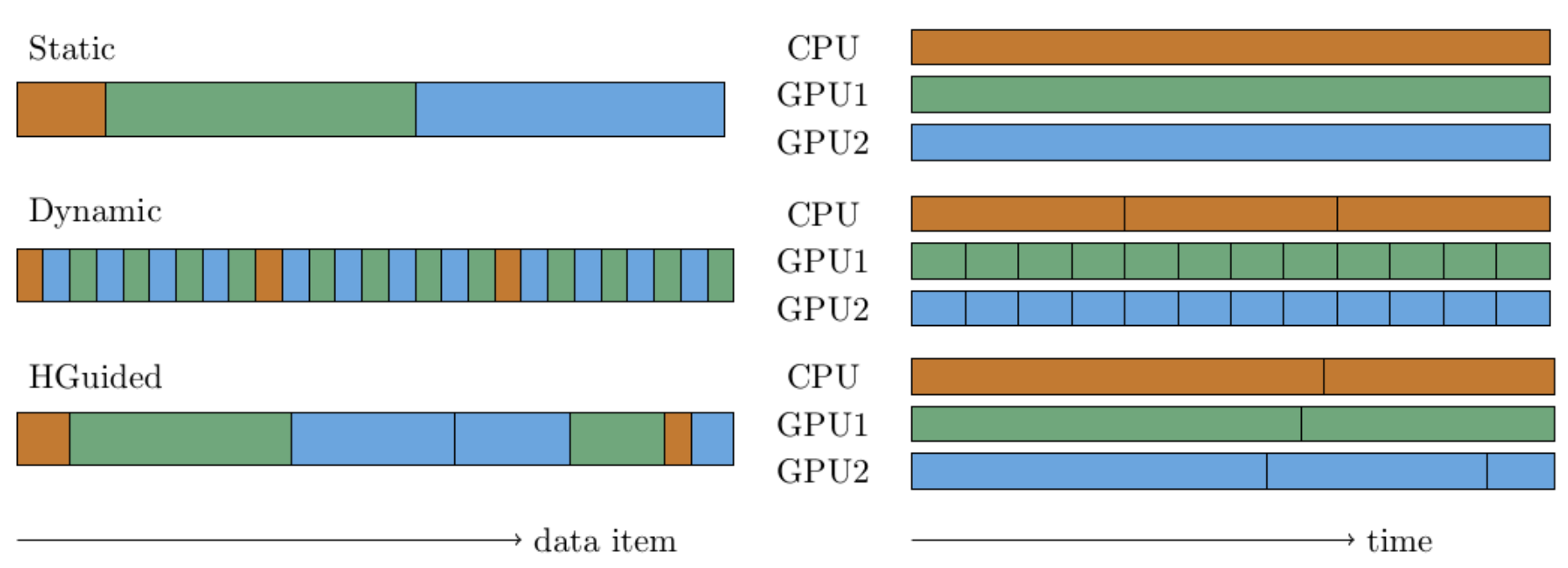
CONTRIBUTIONS: SINGLE SYSTEM IMAGE

- Single System Image.
  - Programmers only see a host and a device.
  - Higher abstraction level of the system,
  - Simplify the programmability
  - Increasing productivity.
- Improves **scalability** and **performance portability**,



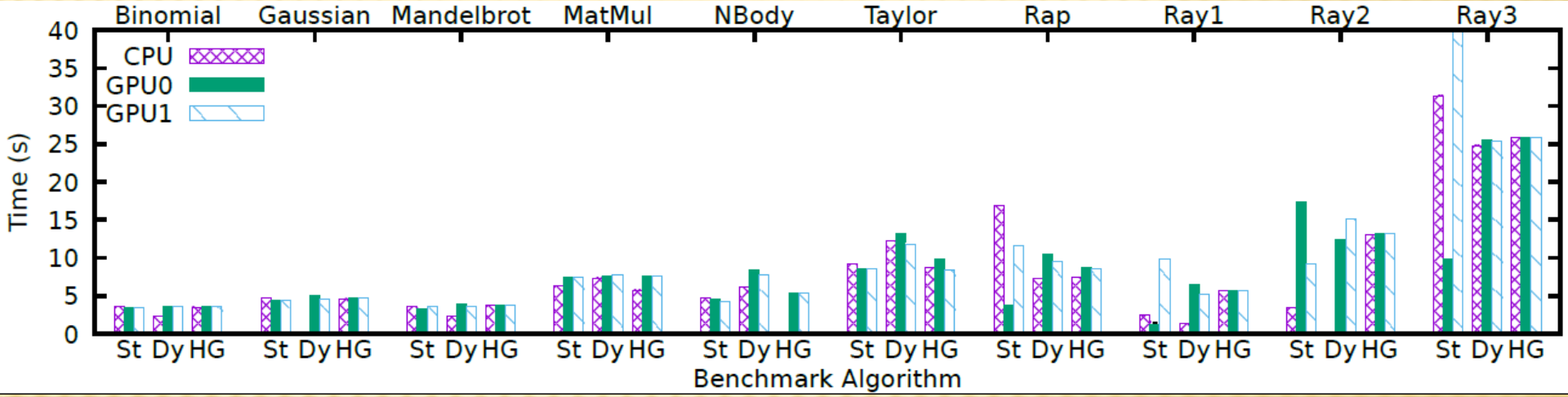
CONTRIBUTIONS: LOAD BALANCING

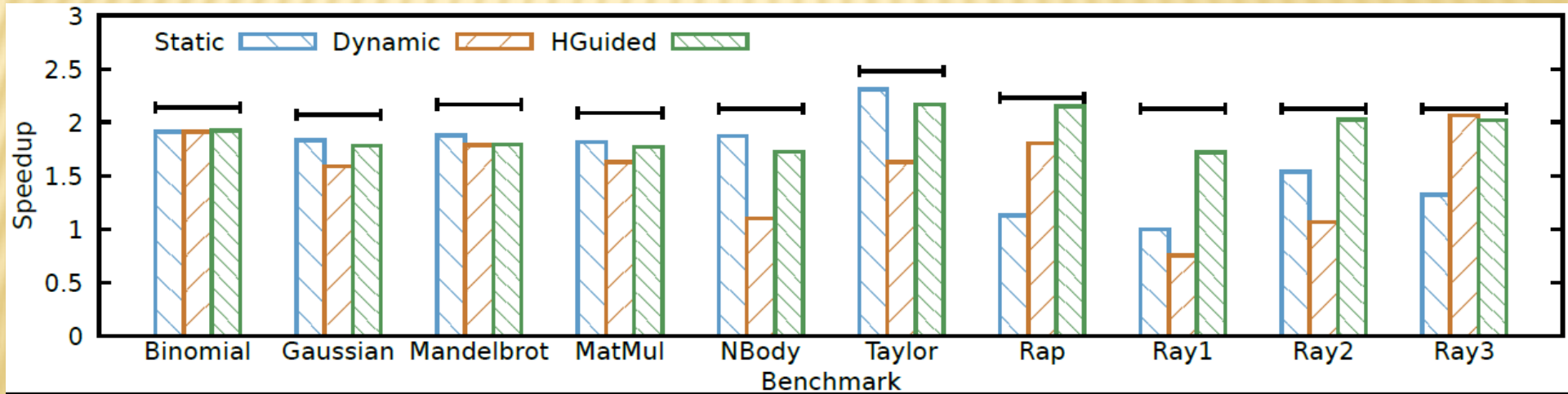
- Set of load balancing models:
  - Static
  - Dynamic
  - Guided
- The programmer can select the most suitable model, depending on the nature of the problem.



EVALUATION: PERFORMANCE

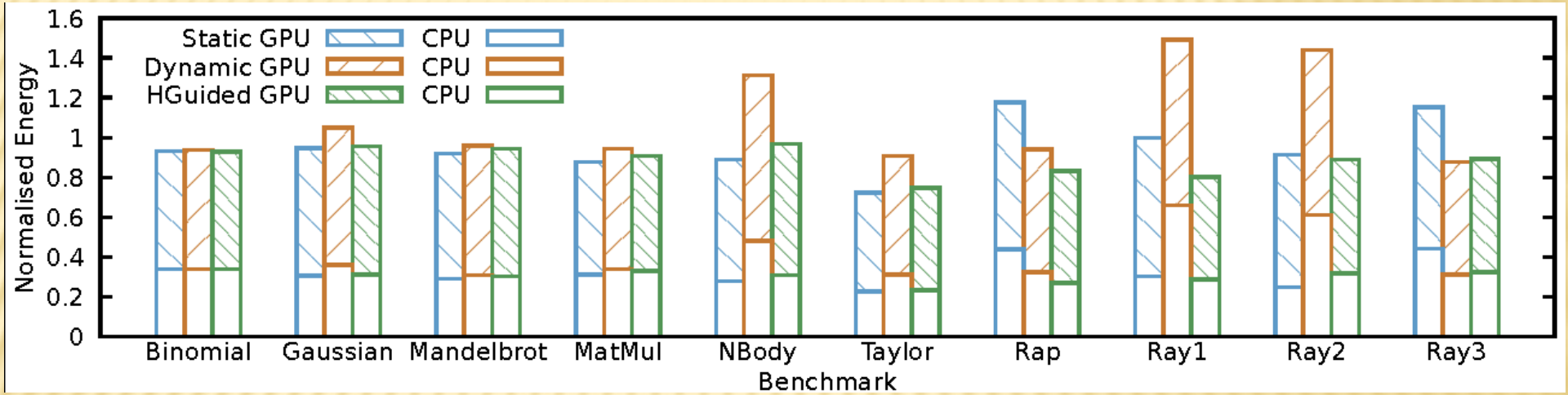
- Experimental Set-up
  - CPU: 2x 2.0 GHz Intel Xeon E5-2620, with 6 cores each
  - GPU: 2x GPU Nvidia K20 with 6 Gbytes of main memory..

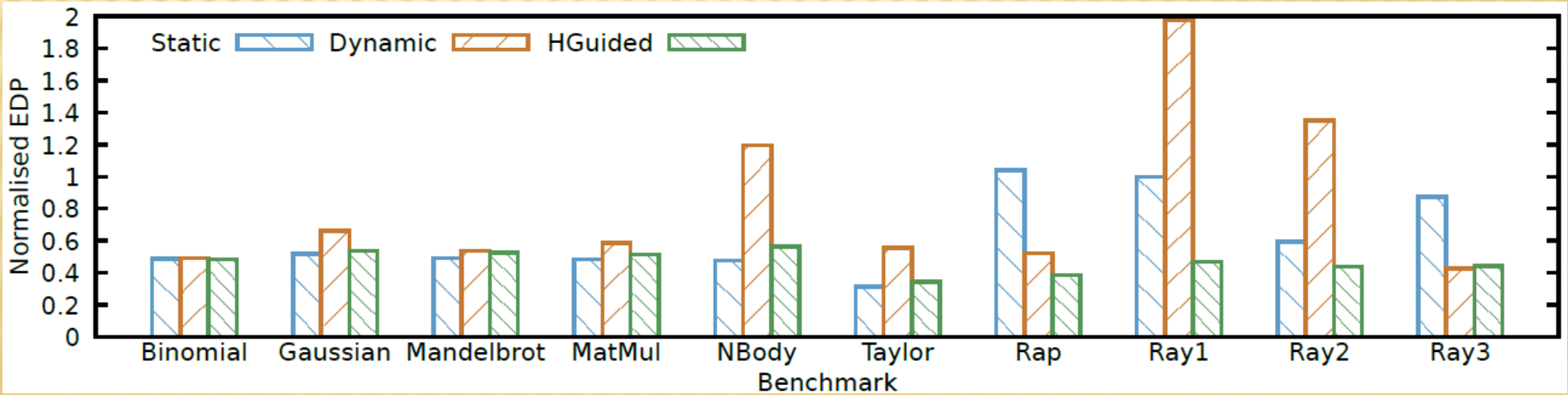




EVALUATION: ENERGY

- **Sauna:** Measuring the energy consumption on CPU and GPU
- Maat more than doubles the energy efficiency of the system





CONCLUSIONS

- The use of the whole heterogeneous system improves the performance and energy consumption.
- At least one algorithm succeeds at adequately balancing the load for every application, with a efficiency over 0.9
- The **Static** approach is the most suitable for regular applications and when the computing powers are known, as it minimises overheads.
  - The other algorithms represent suboptimal yet good solutions.
- For irregular applications the **HGuided** excels and static approaches is strongly unadvisable due to imbalance.
- The **Dynamic** algorithm is a good all-around option when the computing powers of the devices are unknown.