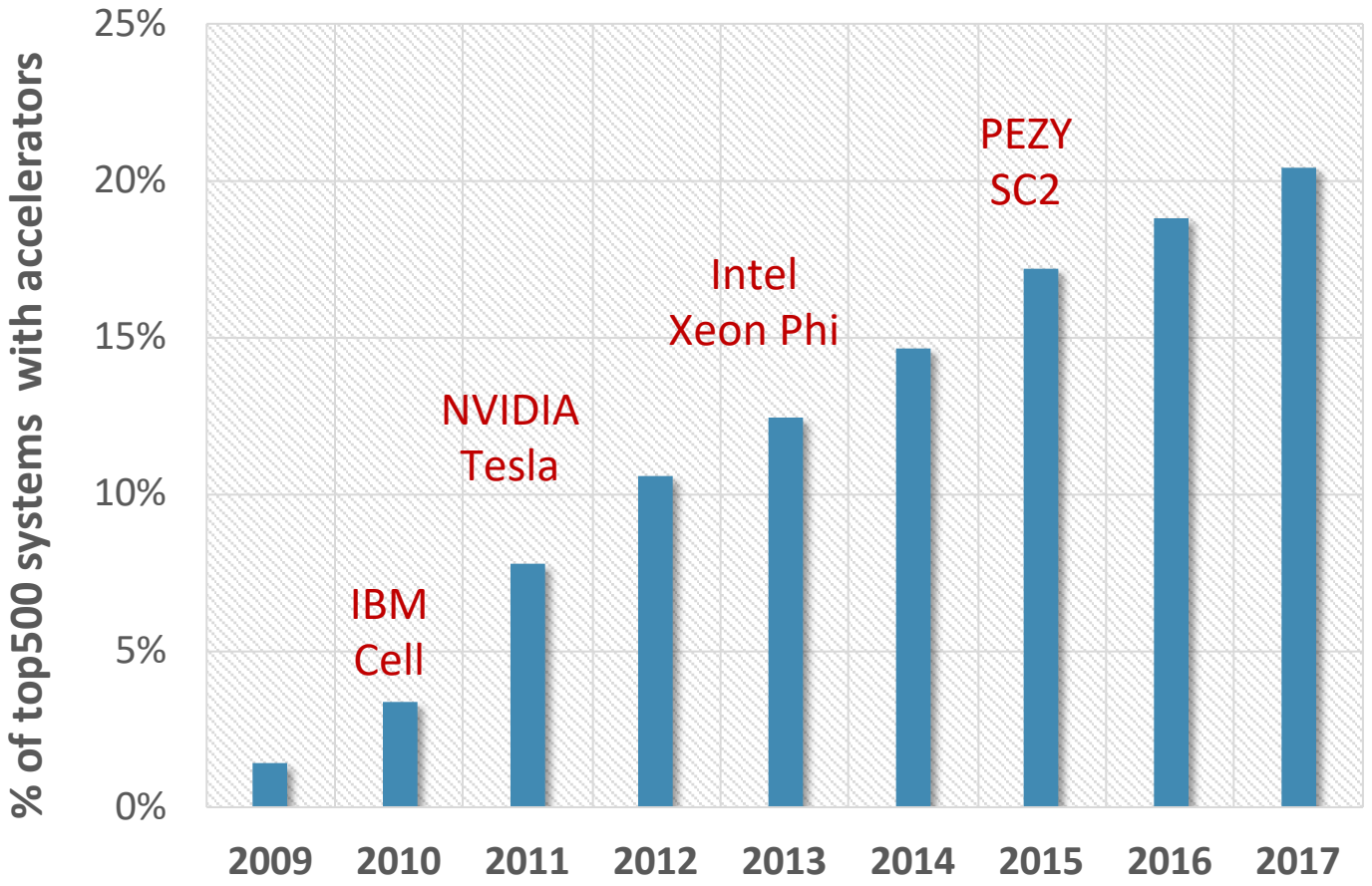
Task Mapping in Soft Heterogeneous Systems

LECTURE
COURSE
TERM

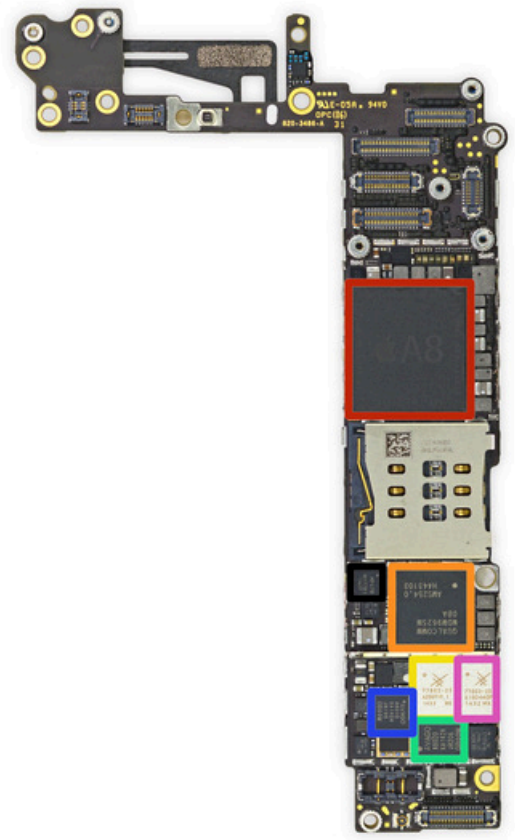
Developed by : Apan Qasem, <apan@txstate.edu>
Work supported by NSF grant OAC-1829644

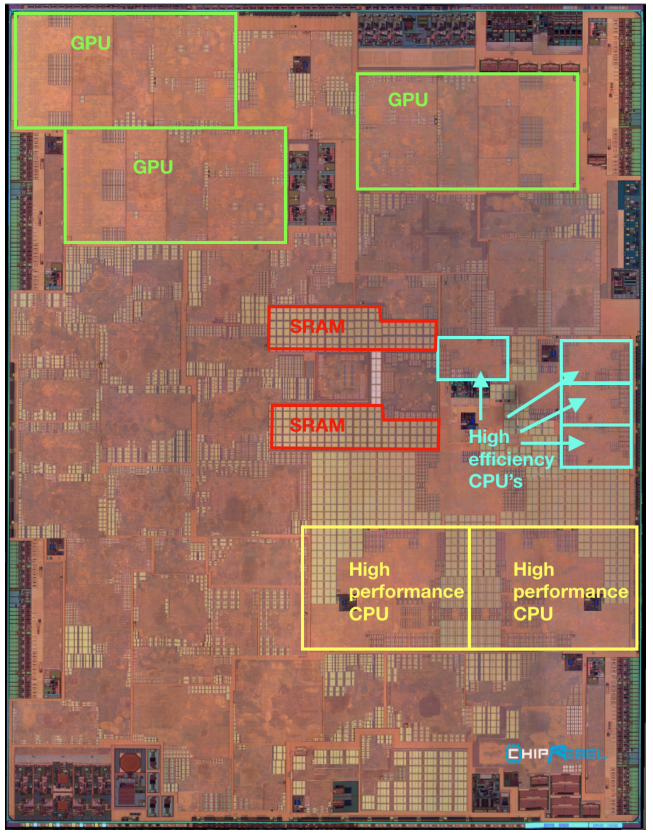
The Rise of Heterogenous Computing



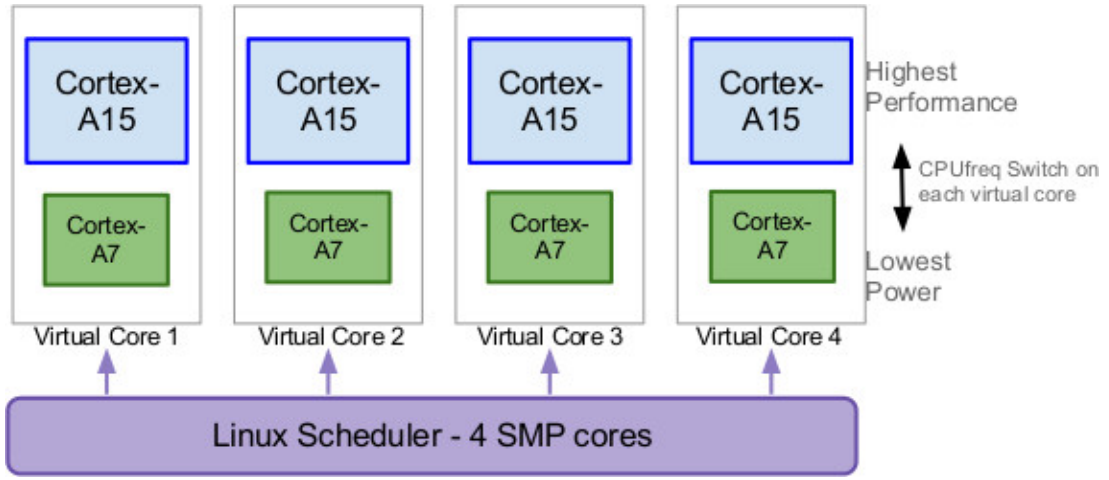
Processor on a Board

Heterogenous Processors





ARM Big-Little Architecture



Computing is not just about performance

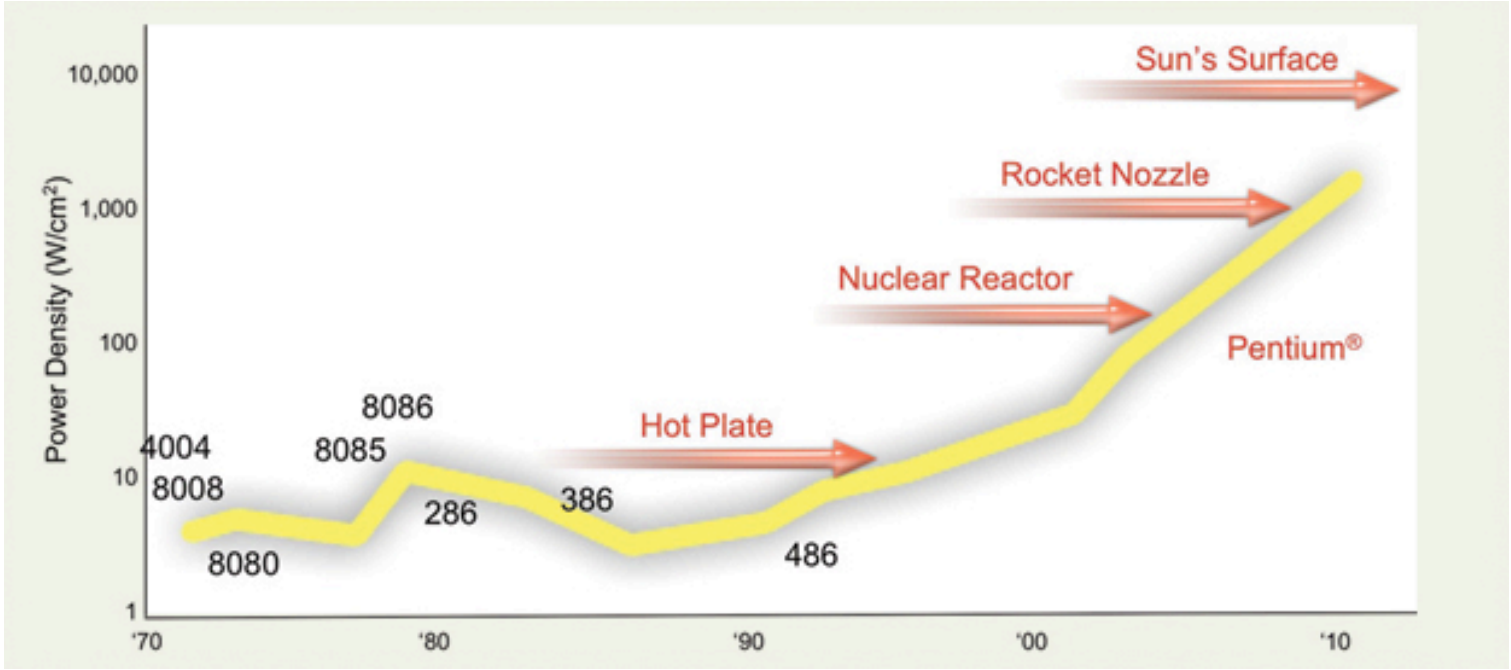
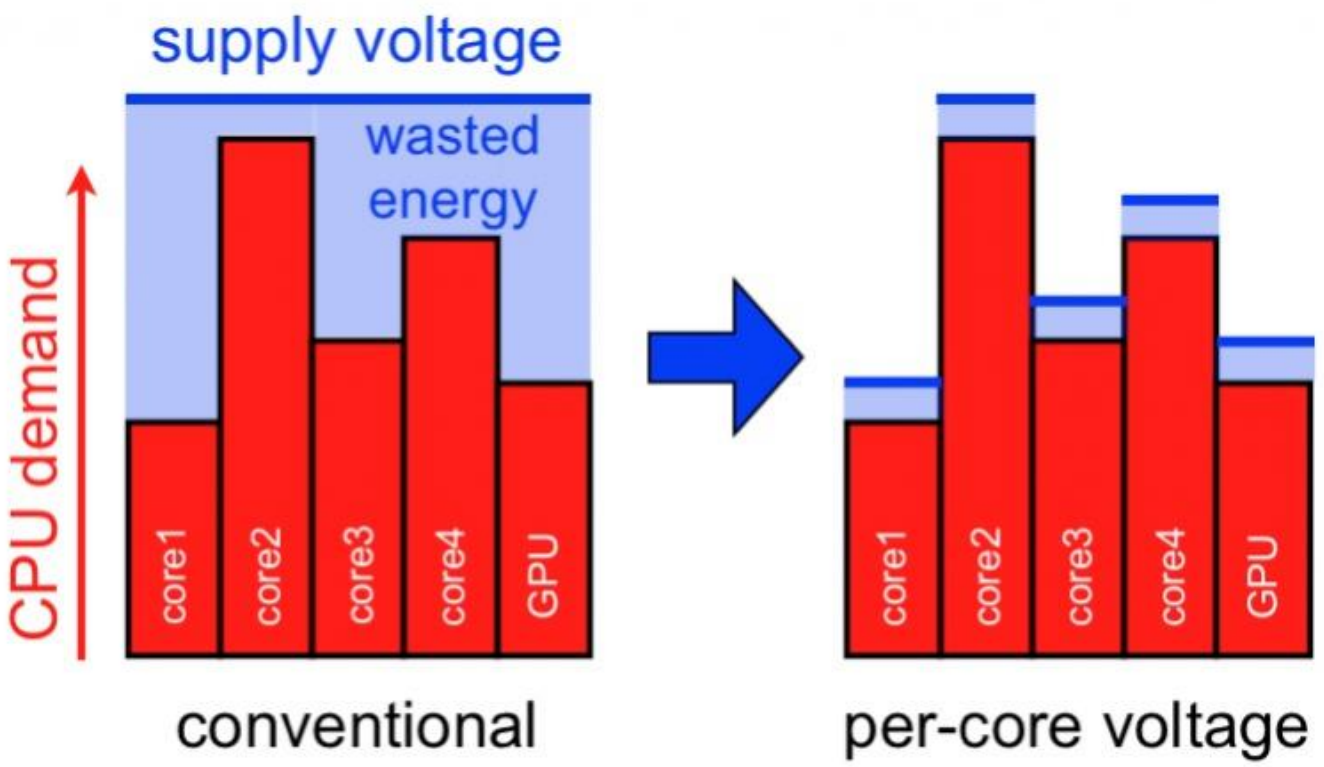


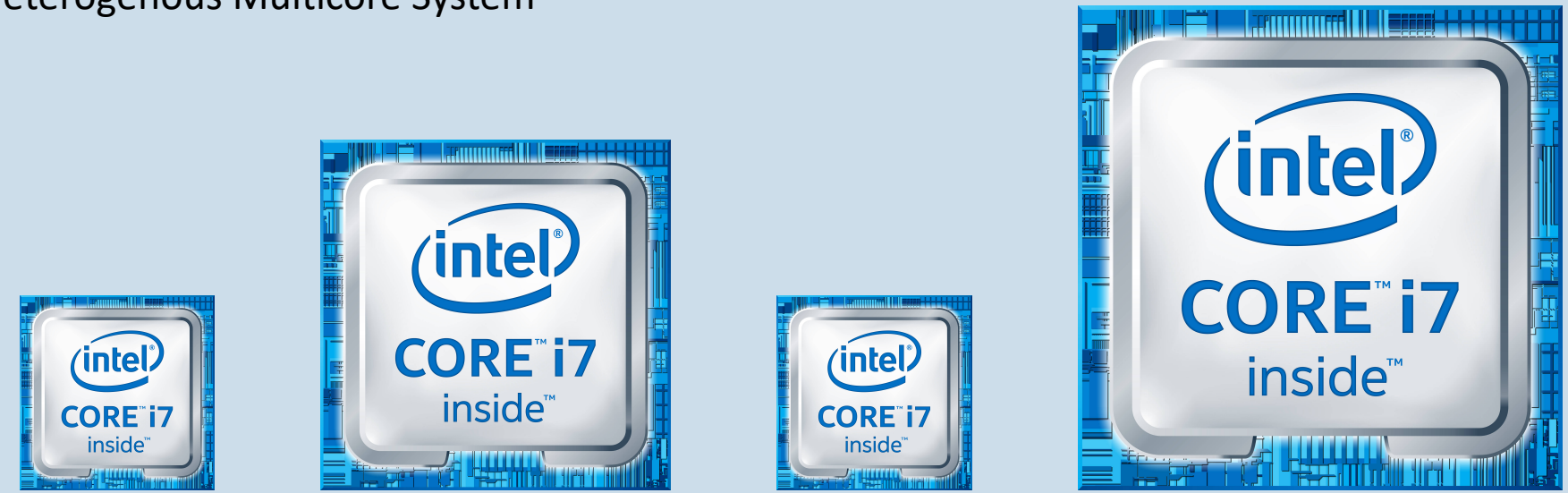
Chart courtesy : Pat Gelsinger, Intel Developer Forum, 2004

DVFS Technology



“Soft” Heterogeneity

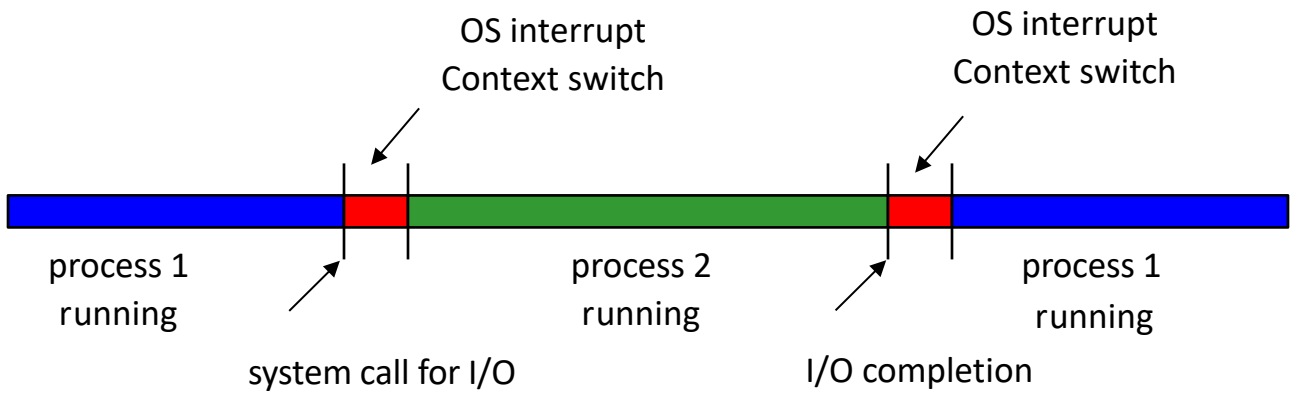
Heterogenous Multicore System



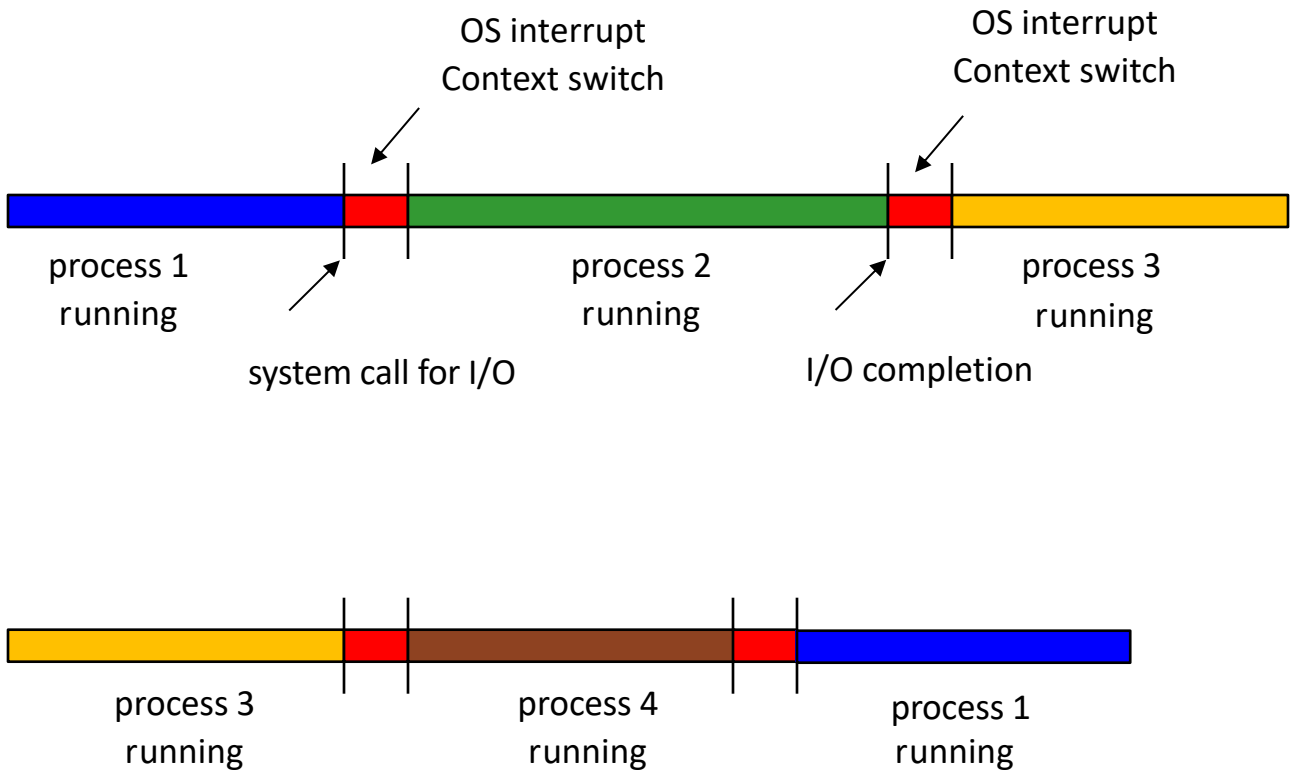
Operational frequency is different

Task Mapping and Scheduling

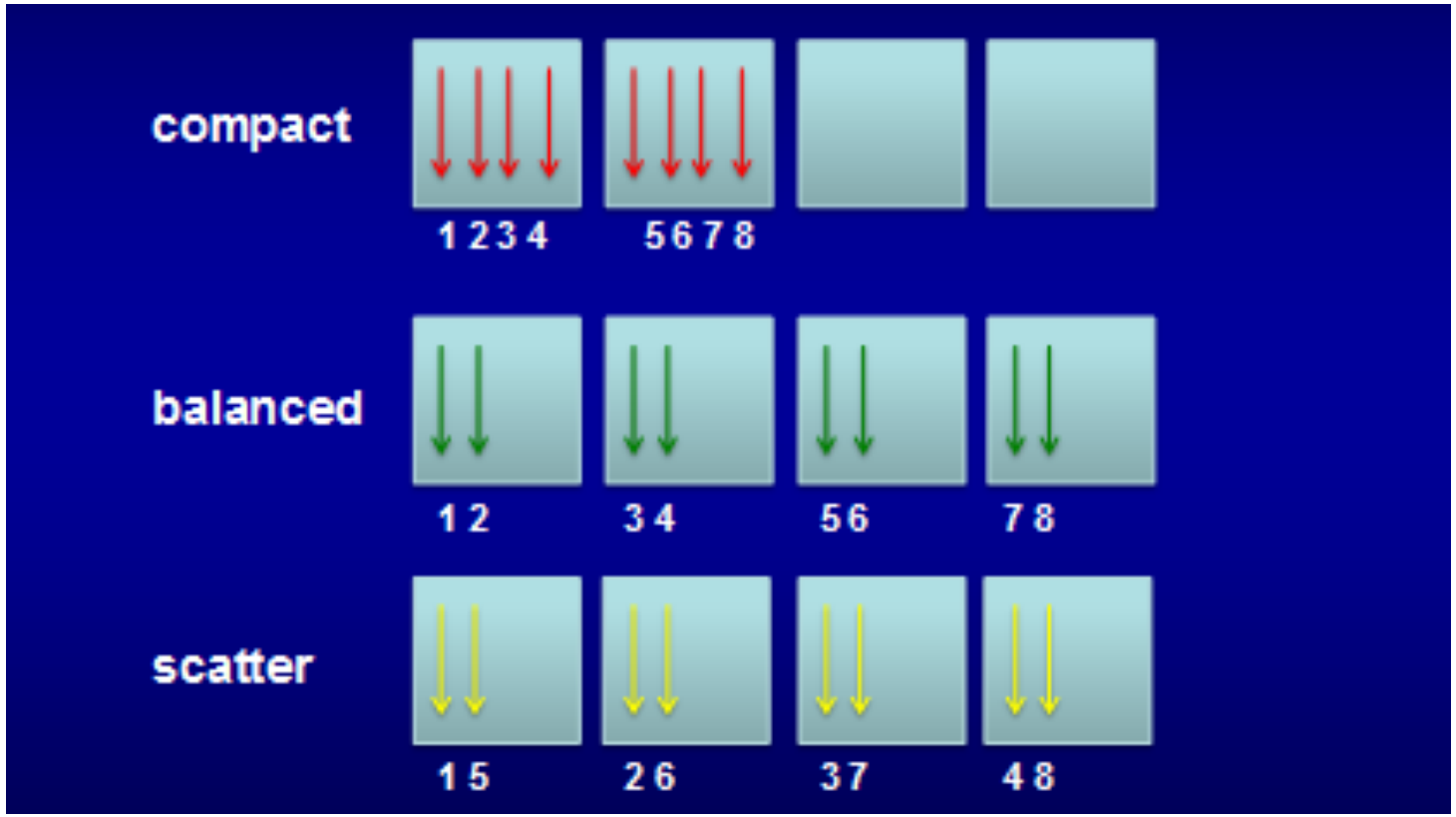
Task Scheduling on Sequential Processors



Task Scheduling on Parallel Processors



Task Mapping and Thread Affinity



Task Mapping on Heterogeneous Systems

- On homogeneous systems, task mapping and scheduling is driven by the workload characteristics and demand
- On heterogeneous systems, the task and scheduling becomes more challenging
- In addition to considering the workload characteristics, the system must factor the capabilities of the processing cores

- Goal is to bind tasks to cores that best matches its requirements
  - If a task is computation-heavy set it's affinity to the core with the highest frequency

Demo: Task Mapping

Reference

image source

- Intel Core i7: intel.com
- ARM big.LITTLE: wikimedia (Open Grid Scheduler / Grid Engine)
- task mapping : Cornell Virtual Workshop (https://cvw.cac.cornell.edu/mic/affinity)