Accelerating a local search algorithm for large instances of the independent task scheduling problem with the GPU

Frederic Pinel    Johnatan Pecero    Pascal Bouvry

Computer Science and Communications
University of Luxembourg

GreenDays@Paris
Outline

Motivation

Initial algorithm

Adaptation

Large instances

Conclusion
Motivation

• Independent tasks
• Makespan, combines several perspectives:
  • User: flowtime
  • Provider: load balance, energy (low machine heterogeneity)
Parallel CGA

Figure: Generating solution
Parallel CGA

- Parallel asynchronous cellular genetic algorithm
- Initialized with heuristic (Min-Min)
- Local search
Feedback
Adaptation

• Simplified algorithm
• Min-Min, incremental formulation
• Increased local search, complete-state formulation
**Results**

Figure: Consistent, high-h. tasks, low-h. machines
Figure: Semi-consistent, high-h. tasks, low-h. machines
Figure: Consistent, low-h. tasks, low-h. machines
Figure: Semi-consistent, low-h. tasks, low-h. machines
Figure: Parallel reduction in Min-Min
Min-Min runtime

Figure: Runtime Min-Min
Performance

Figure: Makespan
Performance

Figure: Runtime
Conclusion

- Failure
- Solution $\rightarrow$ Feedback $\rightarrow$ loop
- Learning process
Machine learning opportunities

- Learn on problem instance
  - Task profiling
  - Co-scheduling
- Learn allocation rules
  - Adapt (parameters, heuristics)
  - Algorithm (oracle: solved instances)
Questions

Thank you.