Methodology for Autotuning of MPI Applications

Antonio Pimenta  Eduardo César  Anna Sikora
{antonio.pimenta, eduardo.cesar, anna.sikora}@uab.cat
Computer Architecture and Operating Systems Department, Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain

ABSTRACT
This paper proposes a methodology designed to tackle the most common problems of MPI parallel programs. By developing a methodology that applies simple steps in a systematic way, we expect to obtain the basis for a successful autotuning approach of MPI applications based on measurements taken from their own execution. As part of the AutoTune project, our work is ultimately aimed at extending Periscope to apply automatic tuning to parallel applications and thus provide a straightforward way of tuning MPI parallel codes. Experimental tests demonstrate that this methodology could lead to significant performance improvements.

MOTIVATION and OBJECTIVES
• MPI is the standard specification used for message passing in parallel distributed applications.
• Inexperienced users will often ignore the tuning parameters, thus using a suboptimal configuration.
• Other methods require a previous study of the architecture and application.
• A general tuning strategy allows flexibility and compatibility with multiple MPI implementations.

We aim to study simple cases of manual tuning on parallel applications to design a methodology that will allow us to perform similar tuning actions automatically. Our main goals are:
1. Identify a set of relevant aspects to optimize.
2. Study the process of manually tuning the application and system and obtain a baseline of performance improvement for a real application.
3. Based on Periscope, develop a viable methodology to apply the process automatically.

OPTIMIZATION ASPECTS
• MPI communication structure:
  • Blocking vs non-blocking
  • Collective vs point to point

• Load balancing:
  • Workload partitioning
  • Distribution strategies

• MPI configuration parameters:
  • Eager limit
  • Buffer sizes
  • Process affinity
  • ...

MANUAL TUNING

- MPI communication structure
  • Changed communication structure
  • Use of non blocking directives
  • Impact: 10-20%
- Load balancing
  • Adaptive redistribution scheme
  • Impact: 40-60%
- MPI configuration parameters
  • Parameter checking
  • Selection of eager limit and buffer size parameters
  • Impact: 0-10%

Communication restructuring has a relevant impact on all 3 workloads.

Load balancing has great impact on performance, due to the initial instability of the application

MPI parameters won’t improve performance due to the fact that they are already optimized in the target system.

REFERENCES

ACKNOWLEDGEMENTS