

КОМБИНИРОВАННЫЕ МОДЕЛИ ПРОГРАММИРОВАНИЯ*Жевжик С.Е.**Донецкий национальный технический университет*

Архитектуры с общей памятью постепенно становятся всё более значимыми, поскольку преимущества данной технологии позволили большему числу центральных процессоров иметь доступ к единому пространству памяти. Кроме того, изготовители всё чаще группируют такие SMP системы вместе, чтобы выйти за пределы одной системы. Коды для передачи сообщений, написанные на MPI, ясно портативны и могут быть легко перенесены на кластерные SMP системы. OpenMP предлагает более эффективную стратегию распараллеливания внутри узла SMP. Следовательно, комбинация общей памяти и парадигмы распараллеливания, основанной на передаче сообщений, в пределах одного и того же приложения может обеспечить более эффективное решение задач, чем чистый MPI. OpenMP и MPI представляют промышленные стандарты для систем с распределённой и общей памятью соответственно [3].

Модель программирования с передачей сообщений (MPI) является моделью с распределённой памятью с внешним контролем параллелизма [1, с.5]. MPI совместим как с распределённой, так и с общей структурой памяти, и позволяет статическое планирование задач. Внешний параллелизм часто обеспечивает большую эффективность, и количество оптимизированных коллективных процедур коммуникации достаточно для достижения оптимальной эффективности. Проблемы с размещением данных наблюдаются редко, синхронизация происходит неявно с вызовами подпрограмм, что естественным образом минимизирует усложнения [2]. Однако MPI имеет некоторые недостатки. Декомпозиция, разработка и отладка приложений могут потребовать значительных временных затрат, и существенные изменения в коде часто необходимы. Коммуникации могут потребовать дополнительных расходов, и кодовая степень детализации часто должна быть большой, чтобы минимизировать временные затраты на вспомогательные операции.

OpenMP – промышленный стандарт для программирования общей памяти [4]. Он основан на комбинации директив компилятора, библиотечных процедур и переменных окружающей среды, что используется для определения параллелизма на машинах совместно используемой памяти. Коммуникация неявна, и OpenMP-приложения относительно легки в имплементации. OpenMP, теоретически, лучше использует архитектуру общей памяти. Коды OpenMP, однако, будут исполняться только на машинах с общей памятью, и концепция размещения данных может стать причиной проблем.

Комбинированный код моделей программирования (MPI и OpenMP) потенциально может предложить наиболее эффективную стратегию распараллеливания для SMP-кластера, а также, позволяя использовать различные особенности обеих парадигм, может обеспечить наилучшую эффективность на отдельно взятом SMP. Большинство комбинированных приложений реализуют иерархическую модель. При разработке комбинированного кода для SMP систем, объединённых в кластер, MPI должен использоваться только для коммуникации между узлами. Имплементация с использованием OpenMP не страдает от проблем балансирования загрузки процессов, и, следовательно, работа кода может быть улучшена.

Данный стиль программирования не будет самым эффективным механизмом для SMP-систем и не может быть расценен как идеальная программная модель для любого кода. В некоторых ситуациях, однако, от имплементации комбинированного подхода может быть получена существенная выгода. Например, в том случае, если параллельный код (MPI) имеет следующие характеристики: плохое масштабирование с процессами MPI из-за несбалансированной загрузки или мелкой зернистости процессов, ограничения памяти из-за использования стратегии копируемых данных, или ограничение на число комбинаций процессов MPI[5]. Кроме того, если система имеет плохо оптимизированную или ограниченную имплементацию масштабирования MPI, применение комбинированного кода также может увеличить эффективность.

Литература

- [1] Антонов А.С. «Параллельное программирование с использованием технологии MPI»: Учебное пособие – М.:Изд-во МГУ, 2004. -71с.
- [2] Воеводин Вл.В. «Технологии параллельного программирования»: Курс лекций - (<http://loi.sccc.ru/lab/hpc/mpi/voevod.htm>)
- [3] Ian Foster. «Designing and Building Parallel Programs» – (<http://www.unix.mcs.anl.gov/dbpp/text/book.html>)
- [4] Лаборатория Параллельных Информационных Технологий. «Что такое OpenMP?» - НИВЦ МГУ – (http://www.parallel.ru/tech/tech_dev/openmp.html).
- [5] Лабусов А.Н. "Технологии распараллеливания" - (<http://www.spbcas.ru/cfd/techn/Parallel.htm>)