

*Старосельский А.Я.*  
*УНК "ИПСА" НТУУ "КПИ"*

## Message Passing Interface

Message Passing Interface (MPI) является наиболее распространённым стандартом интерфейса обмена данными в параллельном программировании, существуют его реализации для большого числа компьютерных платформ. Основным средством коммуникации между процессами в MPI является передача сообщений друг другу. Стандартизацией MPI занимается MPI Forum. В стандарте MPI описан интерфейс передачи сообщений, который должен поддерживаться как на платформе, так и в приложениях пользователя. В настоящее время существует большое количество бесплатных и коммерческих реализаций MPI. Существуют реализации для языков Фортран, С и С++.

Стандартизацией MPI занимается MPI Forum. Последняя версия стандарта – 2.0. В этой версии к MPI добавлены такие важные элементы функциональности, как динамическое управление процессами, односторонние коммуникации (Put/Get), параллельный ввод/вывод. Ожидается появление нового стандарта – 3.0, на данный момент находится в состоянии разработки и утверждения.

И хотя MPI использует 5 и более высокие уровни из модели OSI, дополнения к библиотеке покрывают практически все слои модели OSI. На транспортном уровне MPI использует сокет и протокол TCP. Большинство всех MPI дополнений состоят из специфического набора шаблонов (т.е., программный интерфейс приложения), могущих вызываться из фортрана, С, или С++, и от любого языка, способного к установлению связи с компьютером с такими библиотеками. Преимущества MPI над более старшими библиотеками передачи сообщений – мобильность (поскольку MPI был осуществлен для почти каждой распространяемой архитектуры памяти) и скорость (поскольку каждое выполнение в принципе оптимизируется для технических средств, на которых приложение запущено).

Интерфейс MPI предназначен, чтобы обеспечить необходимую виртуальную топологию, синхронизацию, и функциональность коммуникации, между набором процессов (это проецировалось на узел/сервер/компьютер) в независимом от языка пути, со специфическим для языка синтаксисом, плюс несколько особенностей, которые являются специфическими для языка. Программы MPI всегда работают с процессами, но программисты обычно ссылаются на процессы как процессоры. Обычно, для максимального выполнения, каждому ЦП (или ядро в многопроцессорной машине) будет назначен только один процесс. Это назначение случается во времени выполнения через агента, который запускает программу MPI, который обычно называют к `mpirun` или `mpiexec`.

Архитектуры меняются, с большой внутренней конкуренцией (многопроцессорных систем), лучшим контролем мелкозернистого конкурентного параллелизма (поточность, родство), и более высокими уровнями иерархии памяти. Многопоточные программы могут воспользоваться этими событиями легче, чем обычные приложения. Это уже наложило определенный отпечаток, дополнительные стандарты, для симметрической обработки многих потоков, а именно OPENMPI. Стандарт MPI-2 определяет, как стандартное приложение должно работать в случае с многими потоками, стандарт не требует, чтобы выполнение было многопоточным, или даже безопасным. Несмотря на то, что многопоточные реализации MPI существуют, число многопоточных приложений передачи сообщений мало.