

# Совместное использование стандарта MPI и нитей POSIX для организации обменов сообщениями в кластерных вычислительных системах\*

Е.В. Аксенова, М.Л. Цымблер

Стандарт MPI предназначен для поддержки работы параллельных процессов в терминах передачи сообщений. Стандарт POSIX Threads определяет интерфейс прикладного программиста для управления нитями (легковесными процессами) в рамках процесса, представляющего собой приложение. Данные стандарты могут быть использованы совместно в рамках создания гибридного приложения [1].

Примером приложения, в реализации которого требуется совместное использование стандартов MPI и POSIX Threads, является задача разработки параллельной СУБД для кластерных систем. Ядро параллельной СУБД реализуется как MPI-процесс, запускаемый на узлах кластера. Системные процессы ядра параллельной СУБД реализуются как легковесные процессы, запускаемые в рамках MPI-процесса. Реализация коммуникаций между системными процессами ядра, запущенными на различных узлах кластера, затруднена, поскольку стандарты MPI и POSIX Threads не предоставляют прикладному программисту прозрачный интерфейс обмена сообщениями между нитями, запущенными в рамках MPI-процессов.

Нами предлагается подход к решению данной проблемы, основанный на концепции гибридного процесса. *Гибридный процесс* представляет собой легковесный процесс, запущенный в рамках полновесного процесса. Каждый гибридный процесс имеет свой номер, уникальный в рамках приложения. Передача сообщений между гибридными процессами, выполняющимися на одном узле кластера, реализуется на базе функций стандарта POSIX. Передача сообщений между гибридными процессами, запущенными на различных узлах кластера, реализуется на базе коммуникационных функций стандарта MPI. Данный подход реализован нами в виде библиотеки системных функций, названной MPS (Message Passing System). Интерфейсы основных функций данной библиотеки приведены на Рис. 1.

```
typedef (void*) (*hybrid_process)(void *arglist); // Гибридный процесс
int  MPS_Run(hybrid_process func, void *arglist); // Запустить процесс
int  MPS_Send(void* buf, int length, int dest, int tag); // Отправить сообщение
int  MPS_Recv(void* buf, int length, int src, int tag); // Получить сообщение
```

Рис. 1. Интерфейс основных функций библиотеки передачи сообщений

Данная работа является продолжением работ по исследованию методов передачи сообщений в гибридных приложениях для многопроцессорных вычислительных систем с кластерной архитектурой [2].

## Литература

1. Zhang Z., Savant J., Seidel S. A UPC runtime system based on MPI and POSIX threads // Parallel, Distributed, and Network-Based Processing, 2006. PDP 2006. 14th Euromicro International Conference on Volume, Issue, 15-17 Feb. 2006. –P. 195-202.
2. Аксенова Е.В., Цымблер М.Л. Технология гибридных обменов сообщениями на базе стандартов MPI и OpenMP для кластерных систем // Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ: Труды Всероссийск. науч. конф. (24-29 сентября 2007 г., Новороссийск) –М.: Изд-во МГУ, 2007. –С. 44-47.

\* Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 06-07-89148) и Фонда содействия развитию малых форм предприятий в научно-технической сфере (проект 7434).