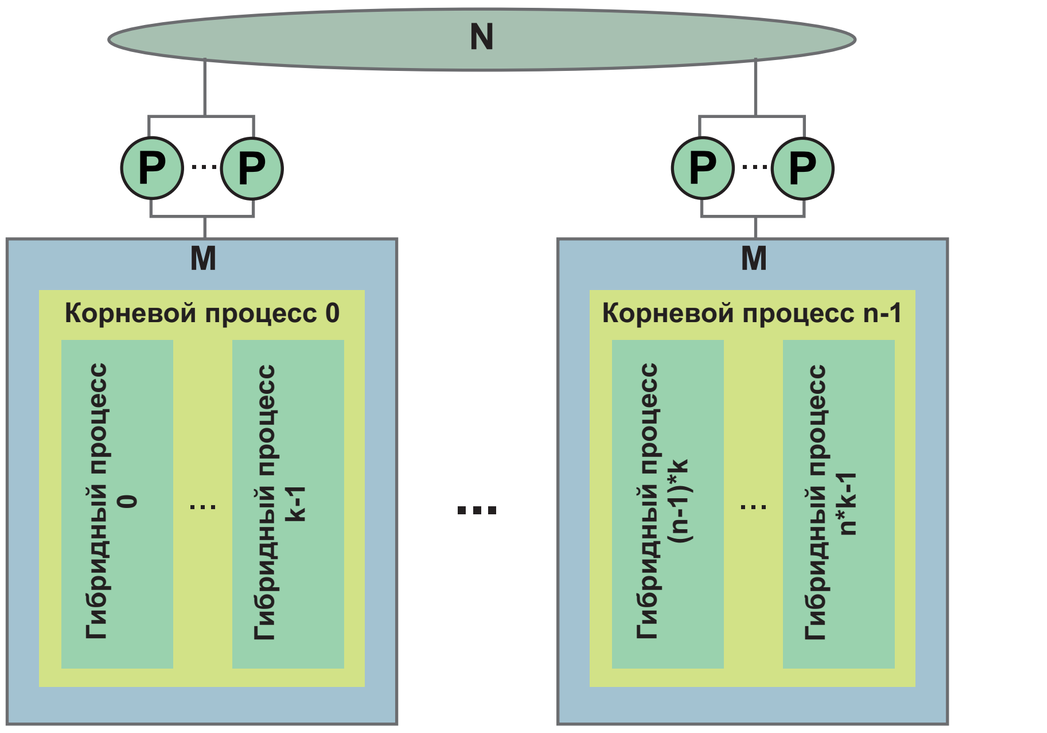
Совместное использование стандарта MPI и нитей POSIX для организации обменов сообщениями в кластерных вычислительных системах*

Е.В. Аксенова, М.Л. Цымблер
evaksen@mail.ru, mzym@susu.ru

Южно-Уральский государственный университет

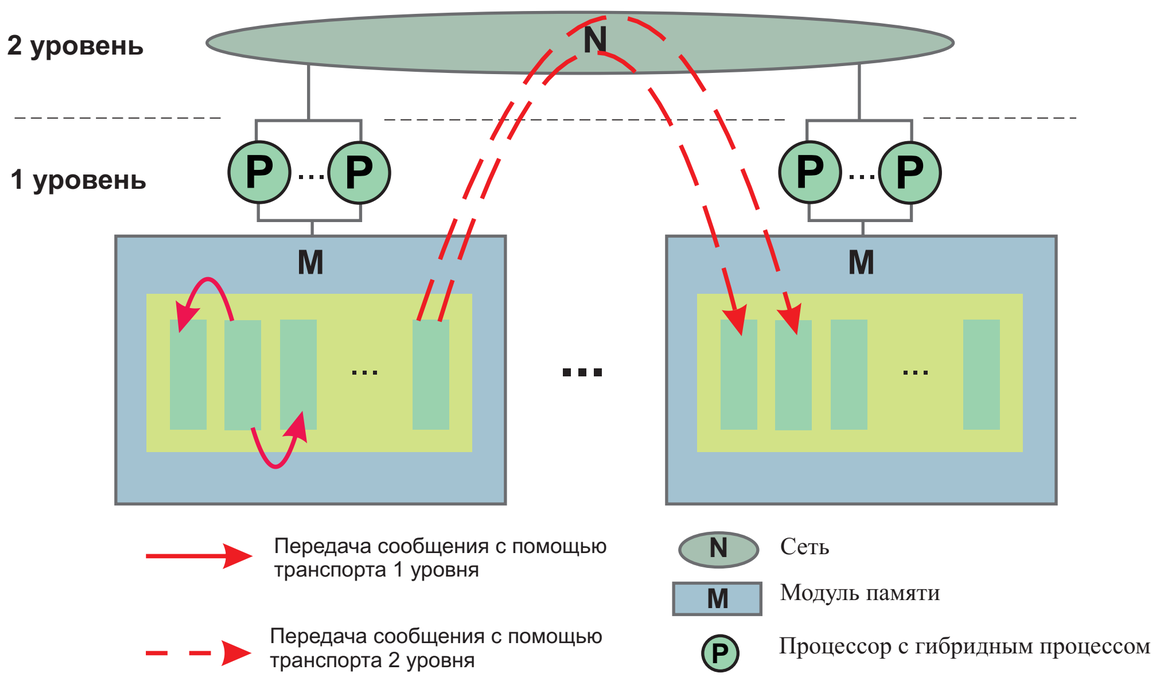
ИЕРАРХИЧЕСКАЯ ПЕРЕДАЧА СООБЩЕНИЙ

Гибридное приложение



Гибридный процесс – легковесный процесс (нить), порожденный корневым процессом.

Технология передачи сообщений в гибридных приложениях



Реализация

Реализация
транспорт 1 уровня - OpenMP, транспорт 2 уровня - MPI

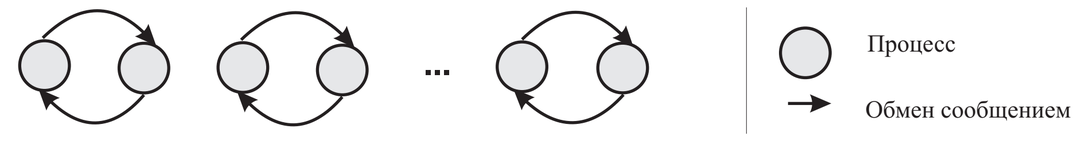
```
int MIX_Init(int* argc, char** argv[]);// Инициализировать
int MIX_Finalize();// Завершить
int MIX_Size(int* size);// Общее количество гибридных процессов
int MIX_Rank(int* rank);// Номер текущего гибридного процесса
int MIX_Send(void* buf, int length, int dest, int tag);// Отправить сообщение
int MIX_Recv(void* buf, int length, int dest, int tag);// Получить сообщение
```

Реализация II
транспорт 1 уровня - POSIX Threads, транспорт 2 уровня - MPI

```
typedef (void*) (*hybrid_process)(void *arglist); // Гибридный процесс
int MPS_Init(int* argc, char** argv[]);// Инициализировать
int MPS_Finalize();// Завершить
int MPS_Run(hybrid_process func, void *arglist);// Запустить гибридный процесс
int MPS_Size(int* size);// Общее количество гибридных процессов
int MPS_Rank(int* rank);// Номер текущего гибридного процесса
int MPS_Send(void* buf, int length, int dest, int tag);// Отправить сообщение
int MPS_Recv(void* buf, int length, int dest, int tag);// Получить сообщение
```

Примеры

Тест Multi-PingPong (пакет Intel ® MPI Benchmarks)

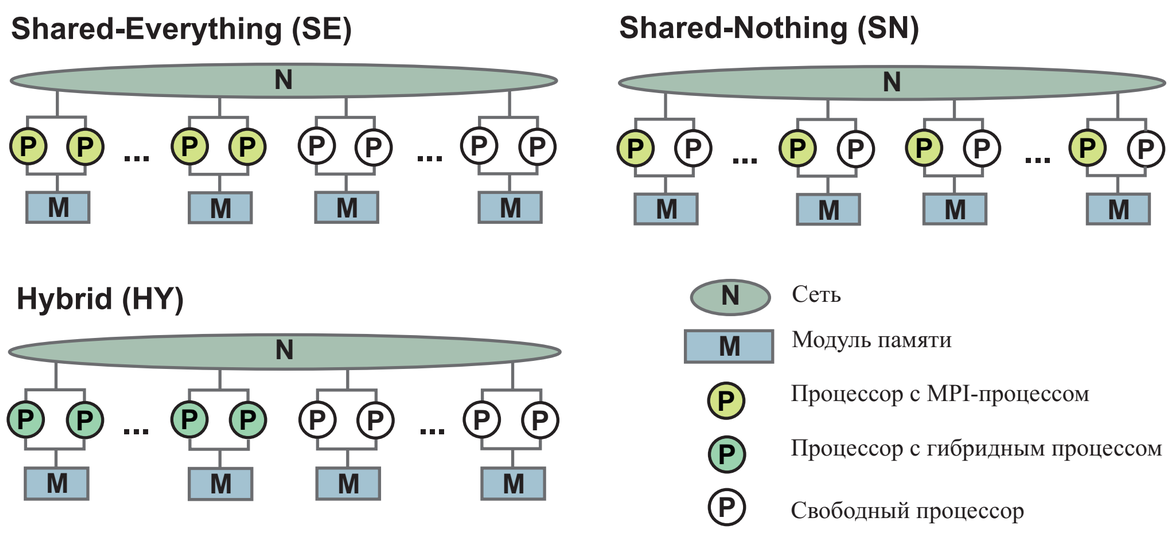



```
/* Тест Multi-PingPong, OpenMP+MPI */
#include "mix.h"
void main(int argc, char *argv[])
{
  MIX_Init(&argc, &argv);                  // Инициализация
  #pragma omp parallel MIX_SHARED_BUF      // Запуск гибридных процессов
  {
    int rank, buf, i;
    MIX_Rank(&rank);
    if (rank%2 == 0) {
      buf = rank;
      MIX_Send(&buf, sizeof(int), rank+1, 0);
      MIX_Recv(&buf, sizeof(int), rank+1, 0);
    } else {
      MIX_Recv(&buf, sizeof(int), rank-1, 0);
      MIX_Send(&buf, sizeof(int), rank-1, 0);
    }
  }
  MIX_Finalize();                          // Завершение
}
```

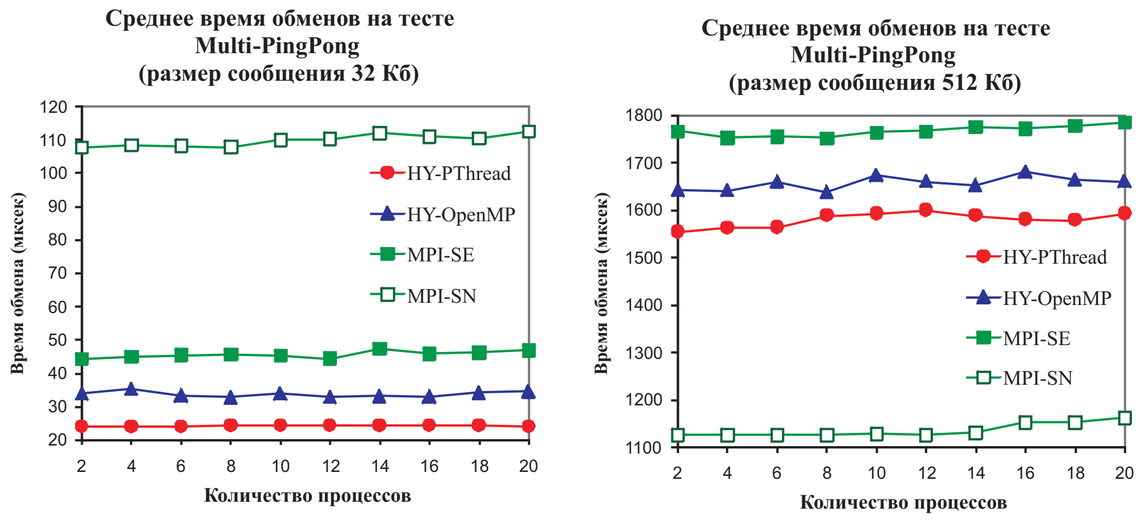
```
/* Тест Multi-PingPong, POSIX Threads+MPI */
#include "mps.h"
void *f(void *);/* Гибридный процесс */
void main(int argc, char *argv[])
{
  int i;
  MPS_Init(&argc, &argv);                  // Инициализация
  for (i=0; i<MPS_NUM_THREADS; I++)        // Запуск гибридных процессов
    MPS_Run(f, NULL);
  MPS_Finalize();                          // Завершение
}
void *f(void *arg)/* Гибридный процесс */
{
  int rank, buf, i;
  MPS_Rank(&rank);
  if (rank%2 == 0) {
    buf = rank;
    MPS_Send(&buf, sizeof(int), rank+1, 0);
    MPS_Recv(&buf, sizeof(int), rank+1, 0);
  } else {
    MPS_Recv(&buf, sizeof(int), rank-1, 0);
    MPS_Send(&buf, sizeof(int), rank-1, 0);
  }
}
```

ЭКСПЕРИМЕНТЫ НА КЛАСТЕРЕ INFINITY

Схемы запуска



Результаты экспериментов на тесте Multi-PingPong



* 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).