

# Семинар 3 (25)

## Программирование сопроцессора Intel Xeon Phi (native mode)

**Михаил Курносов**

E-mail: [mkurnosov@gmail.com](mailto:mkurnosov@gmail.com)

WWW: [www.mkurnosov.net](http://www.mkurnosov.net)

Цикл семинаров «Основы параллельного программирования»  
Институт физики полупроводников им. А. В. Ржанова СО РАН  
Новосибирск, 2016

# Native Mode

- **Запуск программы в Native Mode**

1. Программа компилируется на хост-машине кросс-компилятором (icc): sequential code, OpenMP, Intel Cilk Plus, Intel TBB, Intel MPI
2. Исполняемый файл и все библиотеки копируются в Intel Xeon Phi (scp, NFS)
3. Программа запускается на Intel Xeon Phi (scp, ssh)

- **Ограничения**

- ☐ Объем памяти Intel Xeon Phi (Phi 3120A GDDR5 6 GiB)
- ☐ Отсутствует энергонезависимое хранилище – HDD/SSD  
(по NFS смонтирован каталог /home/micshare – виртуальная сеть через PCI Express)

# Умножение матриц SGEMM (OpenMP)

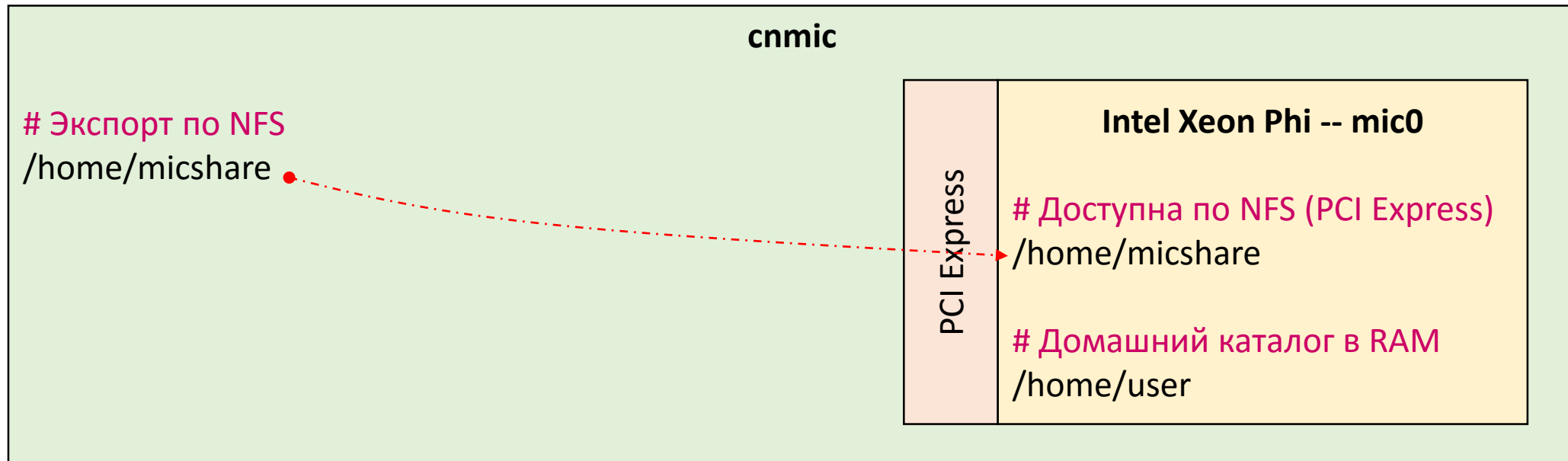
```
/* Matrix multiplication C[n, q] = A[n, m] * B[m, q] */
void sgemm_omp(float *a, float *b, float *c, int n, int m, int q)
{
    #pragma omp parallel
    {
        #pragma omp for
        for (int i = 0; i < n; i++) {
            int k = i * q;
            for (int j = 0; j < q; j++)
                c[k++] = 0.0;
        }
        #pragma omp for
        for (int i = 0; i < n; i++) {
            for (int k = 0; k < m; k++) {
                for (int j = 0; j < q; j++)
                    c[i * q + j] += a[i * m + k] * b[k * q + j];
            }
        }
    }
}
```

# Сборка с ключом -mmic

\$ `icc -mmic -Wall -g -std=c99 -O3 -fopenmp -c sgemm.c -o sgemm.o`

\$ `icc -o sgemm sgemm.o -mmic -lm -fopenmp`

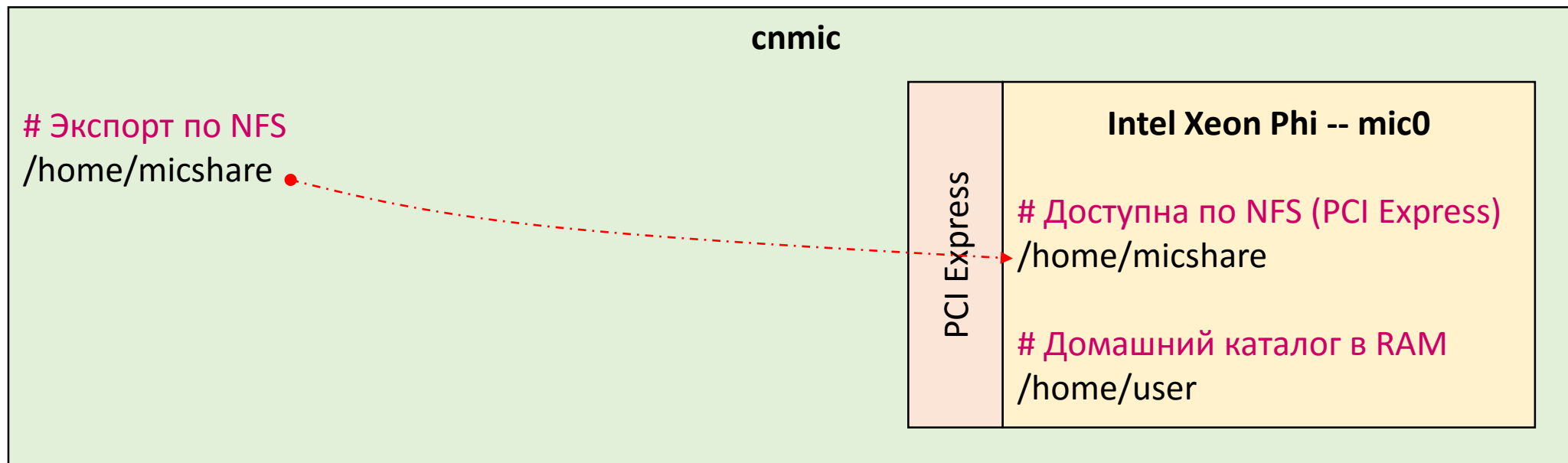
# Копирование исполняемого файла в Intel Xeon Phi



```
# Копирование в домашний каталог
$ scp ./sgemm mic0:
Password:
sgemm                                     100%  50KB  50.1KB/s   00:00

# Запуск на Xeon Phi
$ ssh mic0 ./sgemm
Password:
./sgemm: error while loading shared libraries: libiomp5.so: cannot open shared object file: No such file or directory
```

# Копирование исполняемого файла в Intel Xeon Phi



```
# Копирование и запуск утилитой micnativeloadex
$ cat ./launch.sh
#!/bin/sh

export SINK_LD_LIBRARY_PATH=/home/micshare/libmic
micnativeloadex ./sgemm -d 0 -e "OMP_NUM_THREADS=672 KMP_AFFINITY=explicit,granularity=fine,proclist=[1-224:1]"

$ ./launch.sh
SGEMM N = 1000, M = 1000, Q = 1000, Xeon Phi memory usage 11 MiB (0.19 %)
Phi OMP Native 672 (3 runs): perf 64.11 GFLOPS; time: tavg 0.031198, tmin 0.029922, tmax 0.032041, twarmup 1.061702
```

# Intel Xeon Phi: AVX-512

```
void distance(float *x, float *y, float *z, float *d, int n)
{
    for (int i = 0; i < n; i++) {
        d[i] = sqrtf(x[i] * x[i] + y[i] * y[i] + z[i] * z[i]);
    }
}
```

```
void distance_vec_mic(float *x, float *y, float *z, float *d, int n)
{
    __m512 *xx = (__m512 *)x;
    __m512 *yy = (__m512 *)y;
    __m512 *zz = (__m512 *)z;
    __m512 *dd = (__m512 *)d;

    int k = n / 16;
    for (int i = 0; i < k; i++) {
        __m512 t1 = _mm512_mul_ps(xx[i], xx[i]);
        __m512 t2 = _mm512_mul_ps(yy[i], yy[i]);
        __m512 t3 = _mm512_mul_ps(zz[i], zz[i]);
        t1 = _mm512_add_ps(t1, t2);
        t1 = _mm512_add_ps(t1, t3);
        dd[i] = _mm512_sqrt_ps(t1);
    }
    for (int i = k * 16; i < n; i++) {
        d[i] = sqrtf(x[i] * x[i] + y[i] * y[i] + z[i] * z[i]);
    }
}
```

# OpenMP + SIMD

```
void distance(float *x, float *y, float *z, float *d, int n)
{
    // SIMD only
    #pragma omp simd
    for (int i = 0; i < n; i++) {
        d[i] = sqrtf(x[i] * x[i] + y[i] * y[i] + z[i] * z[i]);
    }
}
```

```
void distance(float *x, float *y, float *z, float *d, int n)
{
    // Threading + SIMD
    #pragma omp parallel for simd
    for (int i = 0; i < n; i++) {
        d[i] = sqrtf(x[i] * x[i] + y[i] * y[i] + z[i] * z[i]);
    }
}
```

# Задание

- Сравните производительность Intel MKL cblas\_sgemm в Native- и Offload-режимах (см. пример в каталоге sgemm\_phi\_native\_mkl)
- Изучите пример particles и предложите реализацию функции distance\_vec:  
OpenMP + AXV-512

