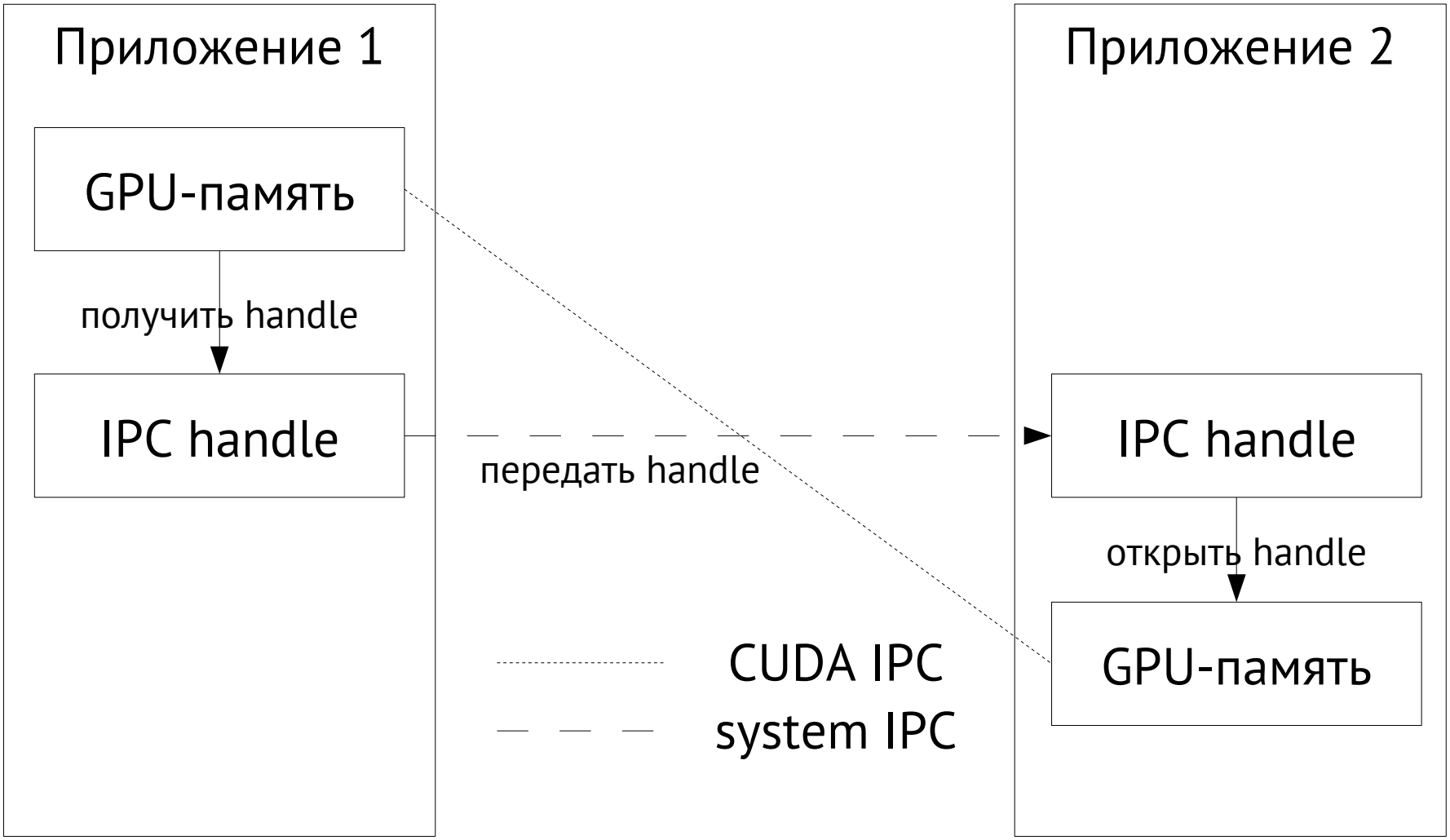
Межпроцессное взаимодействие с CUDA IPC

CUDA IPC

- Совместное использование CUDA-объектов в раздельных приложениях:
  - device memory (напр., double *)
  - CUDA events (cudaEvent_t)
- Работает в рамках одной машины
- Требует «традиционного» IPC для обмена handle'ами

CUDA IPC — схема использования



CUDA IPC – типы и функции (device memory)

- `cudaIpcMemHandle_t`
- `cudaIpcGetMemHandle(cudaIpcMemHandle_t *handle, void *devPtr)`
- `cudaIpcOpenMemHandle(void **devPtr, cudaIpcMemHandle_t handle, unsigned int flags)`
- `cudaIpcCloseMemHandle(void *devPtr)`

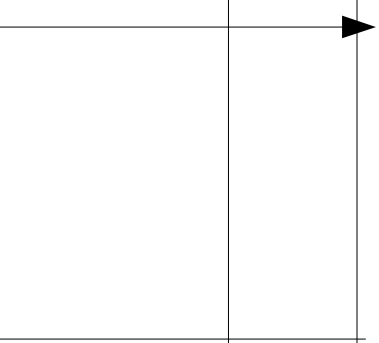
CUDA IPC – функции (device memory)

Приложение 1

```
double *a;
cudaIpcMemHandle_t handle;
cudaMalloc(&a, 1024);
cudaIpcGetMemHandle(&handle, a);
...
<send handle>

<wait>
cudaFree(a);
```

Приложение 2

```
double *a;
cudaIpcMemHandle_t handle;

<receive handle>
cudaIpcOpenMemHandle(&a, handle);
...
kernel<<<...>>>(a);
...
cudaIpcCloseMemHandle(handle);
...
<signal>
```



CUDA IPC – типы и функции (events)

- `cudaIpcEventHandle_t`
- `cudaIpcGetEventHandle(cudaIpcEventHandle_t *handle, cudaEvent_t event)`
- `cudaIpcOpenEventHandle(cudaEvent_t *event, cudaIpcEventHandle_t handle)`

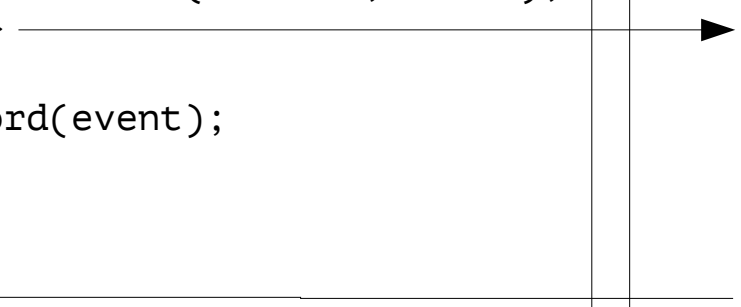
CUDA IPC – функции (events)

Приложение 1

```
cudaEvent_t event;
cudaIpcEventHandle_t handle;

cudaEventCreate(&event,
cudaEventDisableTiming |
cudaEventInterprocess);

cudaIpcGetEventHandle(&handle, event);
<send handle>
...
cudaEventRecord(event);
...


<wait>
cudaEventDestroy(event);
```

Приложение 2

```
cudaEvent_t event;
cudaIpcEventHandle_t handle;

<receive handle>
cudaIpcOpenEventHandle(&event,
handle);
...
cudaEventSynchronize(event);
...
<signal>
```



CUDA IPC – приложения

- MPI
  - MVAPICH – intra-node GPU buffer exchange
- Многокомпонентные архитектуры
  - daemon (C/FORTRAN) + UI (Python, Java)

CUDA IPC: пример из CUDA SDK

- simpleIPC.cu
- использует shared memory:
  - mmap(NULL, …, MAP_SHARED | MAP_ANONYMOUS)
- порождает дочерние процессы:
  - fork(…)
- эмулирует барьерную синхронизацию поверх разделяемой памяти