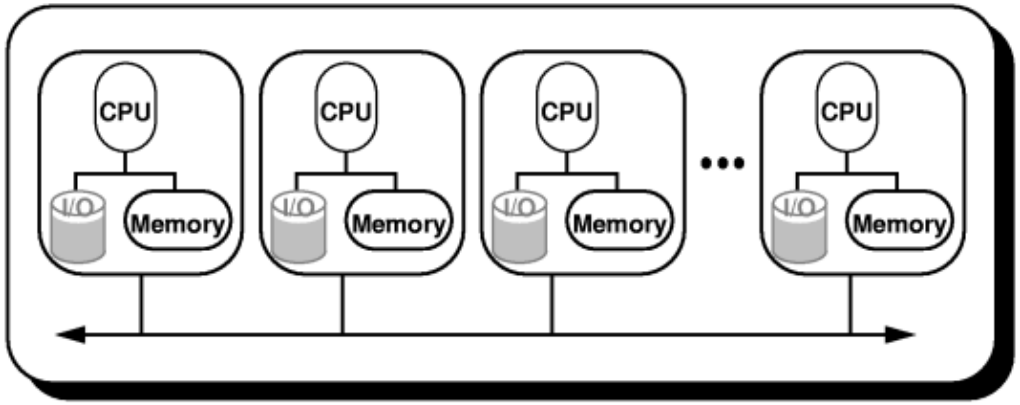
MPI tutoriál (21.3.2011)

MPI
Message Passing Interface

Systémy s distribuovanou pamäťou

- Autonómne procesory s vlastnou pamäťou prepojené komunikačnou sieťou



- Komunikácia realizovaná posielaním správ
- Procesory sú typicky od seba fyzicky vzdialené (clustre, BOINC, „superpočítače“, ...)

MPI (Message Passing Interface)

- API, ktoré umožňuje procesom komunikovať prostredníctvom odosielania a prijímania správ
- používa sa pri vývoji paralelných programov pre superpočítače a clustre
- ciele:
  - použiteľnosť, portabilita, efektívnosť, flexibilnosť
- vznikla v roku 1994 ako výsledok dohody 40 organizácií združených v MPI fórum
- http://www.mcs.anl.gov/research/projects/mpi/

MPI (Message Passing Interface)

- MPI nie je implementácia, resp. nejaká knižnica, je to len špecifikácia API
- štandardizované API umožňuje vývoj portabilných paralelných programov
- výrobcovia hardvéru vytvárajú implementácie MPI optimalizované pre svoj hardvér
- známe implementácie:
  - MPICH, LAM/MPI, WMPI, OpenMPI, Intel MPI
  - MS MPI (optimalizované pre Win HPC Server 2008)

mpiJava

- objektovo-orientované Java rozhranie pre štandardizované MPI (1997)
- http://www.hpjava.org/mpiJava.html
- implementácie:
  - mpiJava – java wrapper (cez JNI) na MPICH implementáciu
  - MPJ Express – čistá Java implementácia
    - http://mpj-express.org/
  - F-MPJ

MPJ Express

- implemetácia java MPI rozhrania len v čistej Jave
  - http://mpj-express.org/
- 2 konfigurácie:
  - multicore konfigurácia
  - cluster konfigurácia
    - cez Java NIO
    - cez Myronet
- ukážka inštalácie...

Ukážka

```
import mpi.*;

public class HelloWorld {
    public static void main(String[] args) {
        MPI.Init(args);
        int size = MPI.COMM_WORLD.Size();
        int me = MPI.COMM_WORLD.Rank();
        System.out.println("Hi from <"+me+">");
        MPI.Finalize();
    }
}
```

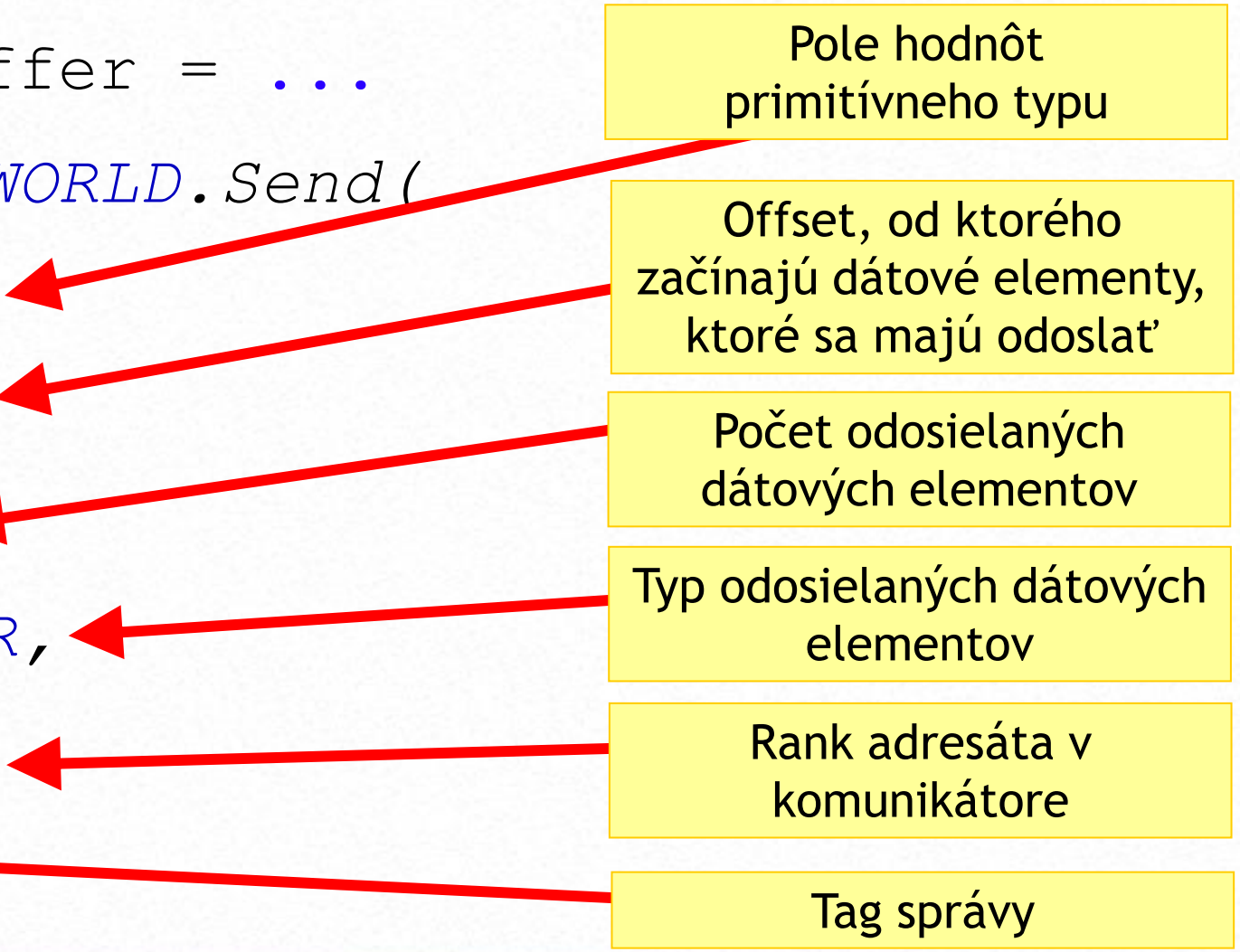
- počet procesov je určený pri spustení programu (argument -np)

Komunikátory a skupiny

- **Skupina** (group)
  - nejaká podmnožina vytvorených procesov
- **Komunikátor** (Comm)
  - umožňuje komunikáciu v rámci skupiny procesov
  - každý proces je v rámci komunikátora identifikovaný svojim rank-om
  - MPI_COMM_WORLD je komunikátor pre skupinu tvorenú všetkými procesmi

Point-to-point komunikácia

```
char[] buffer = ...
MPI.COMM_WORLD.Send(
   buffer,
   offset,
   count,
   MPI.CHAR,
   target,
   tag);
```



Mapovanie typov elementov

MPI datatype	Java datatype
MPI.BYTE	byte
MPI.CHAR	char
MPI.SHORT	short
MPI.BOOLEAN	boolean
MPI.INT	int
MPI.LONG	long
MPI.FLOAT	float
MPI.DOUBLE	double
MPI.PACKED	

Príjem správy

```
Status Comm.Recv(Object buf,
    int offset,
    int count,
    Datatype datatype,
    int source,
    int tag)
```

Prijímací buffer je určený ako časť (určená offsetom a parametrom count) poľa hodnôt primitívneho typu

Rank odosielateľa (alebo MPI.SOURCE_ANY)

Tag prijímanej správy (alebo MPI.TAG_ANY)

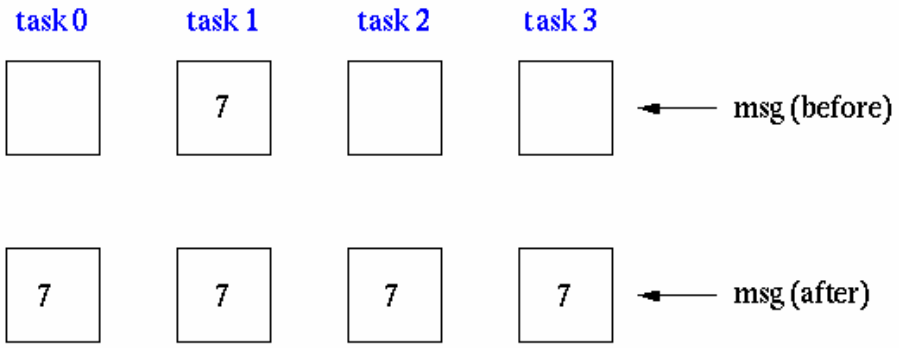
Zo Status objektu sa vieme dozvedieť skutočný počet prijatých dátových elementov, odosielateľa a tag správy.

Typy point-to-point komunikácie

- blokované vs. neblokované
  - pri blokovaných operáciach sa čaká na dokončenie operácie (napr. pri odosielaní sa čaká, kým správa bude odoslaná; po ukončení blokovanej operácie je možné znovupoužívať buffre) … (napr. MPI_Isend)
- synchrónne vs. asynchrónne
  - pri synchrónnych operáciach máme potvrdenie prijatia správy príjemcom (napr. MPI_Ssend)
- Send a Recv sú asynchrónne blokované.

Kolektívna komunikácia (1)

- MPI_Barrier (Barrier)
  - blokuje vykonávanie procesu dokiaľ všetky procesy v komunikátore nevykonajú túto operáciu
- MPI_Bcast (Bcast)
  - pošle správu všetkým procesom v komunikátore, resp. prijme broadcastovanú správu



Kolektívna komunikácia (2)

- Scatter
- Gather
- Allgather
- Reduce
- Allreduce
- Alltoall
- Scan

Ďalšie charakteristiky

- MPI umožňuje vytvárať vlastné skupiny z existujúcich skupín
  - ku každej skupine možno vytvoriť komunikátor, ktorý je užitočný v prípade kolektívnej komunikácie
- MPI umožňuje definovať virtuálne topológie procesov
  - mriežka, graf
  - MPI implementácia môže túto informáciu využiť na mapovanie procesov na fyzické procesory

Ďakujem za pozornosť !