**MPI Cheat Sheet** Arguments from the main function Send from buffer b, c elements of data type d to rank r. The communication is marked with tag t. The function is blocking, b can safely be used after it but data may not have yet been delivered. Called at the start of any MPI program int MPI\_Send( ↓ void \*b, ↓ int c, ↓ MPI\_Datatype d, ↓ int reiceiver, ↓ int t, ↓ MPI\_Comm) int MPI Init( 1 int \*, 1 char \*\*\*) num el(v) MPI INT [0, num tasks) [0, ...) MPI\_COMM\_WORLD &argc &argv &v[3] MPI CHAR NULL NULL &a MPI\_FLOAT v+5 MPI LONG Called at the end of any MPI program AUTOMATIC CONTROL AND COMPUTERS Receive in buffer b, c elements of data type d from rank r. The communication is marked with tag t. int MPI Finalize() The function is bloking, b can be safely used and the data was delivered. Gives the number of tasks int MPI\_Recv(↑void \*b, ↓int c, ↓ MPI\_Datatype d, ↓ int sender, ↓int t, ↓ MPI\_Comm, ↑ MPI\_Status \*) int MPI Comm size ( ↓ MPI Comm, ↑ int \* ) num\_el(v) MPI INT [0, num tasks) MPI COMM WORLD &v[3] [0,..)MPI CHAR MPI ANY SOURCE MPI\_COMM\_WORLD &Stat MPI\_FLOAT [ 0, .. ) MPI\_STATUS\_IGNORE v+5 MPI LONG Made by Cristian Chilipirea &num\_tasks Stat.MPI SOURCE, Stat.MPI TAG MPI\_ANY\_TAG Gives the id (rank) of the current (calling) task Sends (Broadcasts) c elements of data type d from buffer b from rank r to all other tasks in buffer b. int MPI Comm rank (↓ MPI Comm, ↑ int \*) All tasks have to call this function with the same value for root. Rank 0 Rank 2 Rank 3 Rank 1 MPI\_COMM\_WORLD int MPI Bcast (\$\partial void \*b, \$\psi\$ int c, \$\psi\$ MPI Datatype d, \$\partial \text{int root}\$, \$\psi\$ MPI Comm ) 6 8 before num el(v) MPI INT MPI COMM WORLD &rank MPI CHAR &v[3] [0,..)[ 0, num\_tasks ) c == 2root == 1 Synchronizez all tasks at the call of the barrier &a MPI FLOAT v+5 MPI LONG 6 8 6 8 6 8 6 after int MPI Barrier ( \( \square\) MPI Comm comm ) MPI COMM WORLD Splits the elements from sb of datatype sd on rank root in num tasks chunks of size sc. Rank 0 Rank 1 Rank 2 Rank 3 Every task receives its appropriate chunk in rb. For simplicity sc == rc, sd == rd. All tasks have to call this function with the same value for root. 8 5 2 9 3 6 before int MPI Scatter ( ↓ void \*sb, ↓ int sc, ↓ MPI Datatype sd, ↑ void \*rb, ↓ int rc, ↓ MPI Datatype rd, ↓ int root, ↓ MPI Comm) sc == 2root == 1rc == sc v num\_el(v)/num\_tasks MPI INT MPI COMM WORLD v num el(v)/num tasks MPI INT &v[3] [0,...) MPI CHAR &v[3] [0,...)MPI CHAR 5 3 [0, num tasks) MPI FLOAT MPI\_FLOAT &a &a MPI LONG v+5 MPI LONG Gathers sc elements from all sb of datatype sd on all tasks and places the num\_tasks chunks of size rc in rb on task of rank root. Every task sends its appropriate chunk in rb. For simplicity sc == rc, sd == rd. Rank 0 Rank 1 Rank 2 Rank 3 All tasks have to call this function with the same value for root. 3 6 8 before int MPI Gather ( ↓ void \*sb, ↓ int sc, ↓ MPI Datatype sd, ↑ void \*rb, ↓ int rc, ↓ MPI Datatype rd, ↓ int root, ↓ MPI Comm) v num\_el(v)/num\_tasks MPI\_INT MPI\_COMM\_WORLD sc == 2 root == 1 v num el(v)/num tasks MPI INT rc == sc [0...) MPI CHAR &v[3] MPI CHAR &v[3] [ 0, num\_tasks ) &a MPI\_FLOAT &a MPI FLOAT 5 2 9 v+5 MPI LONG MPI LONG v+5