partition nodes fractional {0.8,0.2}

-------- Version Propagation II---------------

Context Chain

data structure

node

context

opened yes/no

list of “node” s

operations:

construct

send

receive

update/insert

isComplete ?

getNextOf(node)

isFinal(node)

task

“context id” field is needed

“end-of-context” task type

listener

three legs :

creator,host,requester

“context id” field is needed

send

version upgrade algorithm:

for any data owned by Node x

if there is no more Listener(read) in current context

if there is no more Task(write) in current context

if end-of-context task is received

if next context is determined

upgrade version

global context new methhods:

host-list = whoAreIn()

host-list = whoAreOut()

bindHosts(me,dest)

host = pairHost(me)

filters/gates

context-host-filter

-------------------------------------------

processor group structure

Context host policy

to be moved inside the MatAsm and Cholesky

not global

set active host policy(CHP::….)

set group count ( a,b,c)

set host partitions (100%, {a%,1-a%} , {b%,c%,1-b-c} )

Context fan in/out > 1 ?

Clone input data for other context, after gen tasks finished remove it

Kernels

Link to Shared Mem Libraries:

SuperGlue

BLAS,LAPACK

Fortran interface

dynamic library load

Dynamic load balancing

Iterative tasks

Fault tolerance

Sparse Matrix

Recursive Tasks

Dynamic Task Generation

Events Actions

**Data**

*Received*:

For all data->listeners :

If (version matches)remove lsnr

For all tasks:

If (data matches)

If (version matches)

Task.data.state = ready

Add task to run-check queue

Raise data-ready event

*Ready*:

For data->listeners:

If (version matches) lsnr.send\_data

Check tasks for run

*Sent to host ‘h’*:

Remove data->listeners

**Listener**

*Received*:

Add to list

Attach to corresponding data

If (data ready) lsnr.send\_data

*Sent*:

Remove from list

**Task**

*Received:*

Add to list

Check task for run

*Sent:*

Remove from list

*Ready:*

Call kernel

*Finished:*

Upgrade out-data version

Remove from list

Actions  Events

**Data**

*Upgrade version*:

‘Data ready’

**Listener**

*Remove:*

If (me == listener.host)

Upgrade data version

**Task**

*Check for run*

‘task ready’

*Call kernel*

‘task finished’