Groups

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | | |

****

****

My groups

Home

Starred

Favorites

Click on a group’s star icon to add it to your favorites

Recently viewed

Recent searches

timeout (in littleproxy)

Recently posted to

MV5sim

[Privacy](https://www.google.com/intl/en/policies/privacy/) - [Terms of Service](https://www.google.com/intl/en/policies/terms/)

|  |  |
| --- | --- |
| gem5-gpu Developers List › |  |
|  |  |
| CPU/GPU ID?  13 posts by 4 authors |  |

|  |  |  |
| --- | --- | --- |
|  | **Jintack Lim** | 4/12/14 |

Hi,

I'm working on the memory system, RubyMemoryController.cc.

For some reason, I need CPU/GPU ID which made memory request.

Could you tell me how I can do it?

Jintack

|  |  |  |
| --- | --- | --- |
|  | **Joel Hestness** | 4/13/14 |

**Other recipients:** incredi...@gmail.com

Hi Jintack,

  You would need to modify a few things to pass this information with the memory access:

  1) In the packet class (gem5/src/mem/packet.\*), you'll need add fields to track the core type/ID as necessary

  2) In the RubyRequest (gem5/src/mem/protocol/RubySlicc\_Types.sm), you'll need to add analogous fields

  3) In the cores that are sending memory accesses (e.g. gem5-gpu/src/gpu/shader\_lsq.\*), you'll need to set the packet fields as appropriate

  4) In the Ruby cache sequencer (gem5/src/mem/ruby/Sequencer.\*), when the RubyRequest is created, you'll need to copy the packet fields over to their analogous RubyRequest fields

  Beyond this, it looks like you'll need to track where these RubyRequests flow through the cache protocol that you're working with (e.g. VI\_hammer) in order to make sure that the ID bits are added to messages between cache controllers and the directory, and then down to the memory controller.

  Hope this is enough info to get a jump on it.  Let me know if you have further questions,

  Joel

- show quoted text -

--

  Joel Hestness

  PhD Student, Computer Architecture

  Dept. of Computer Science, University of Wisconsin - Madison

<http://pages.cs.wisc.edu/~hestness/>

|  |  |  |
| --- | --- | --- |
|  | **Jintack Lim** | 4/16/14 |

**Other recipients:** incredi...@gmail.com

Thanks Joel.

Now, I'm at step 4).

I can check coreID and coreType at Sequecner.cc.

I added coreID and coreType to Packet class.

The big picture I have is to convert

Packet -->  RubyRequest --> MemoryMsg.

However, I have trouble to go through cache controllers and down to the memory controller.

1. In Sequencer::issueRequest function,

packet pointer is already saved to RubyRequest class, pkt member variable.

However, when I check the pkt value using gdb in L1Cache\_Controller::wakeup:467, it is "0".

I also made a new member variable in RubyRequest class, however the value of that variable is not passed correctly.

I guess both of pkt and my own variable are not used currently, so they may be optimized out.

2. In RubyController::enqueue function,

the data passed from upper layer is "MemoryMsg" object.

Where is the conversion from RubyRequest to MemoryMsg happening?

3. I did not do step 2) you mentioned.

This is the only place I can guess.

RubySlicc\_Exports.sm:external\_type(**Packet**, primitive="yes");

Thank you!

- show quoted text -

|  |  |  |
| --- | --- | --- |
|  | **Matt Poremba** | 4/16/14 |

Hi Jintack,

I did something similar a few years back (for CPU-only gem5/ruby), maybe I can provide more details about the request conversions.

The C++ code generated by SLICC allocates a new request at each cache/directory level, so you will need to copy the values at each cache level.

The big picture in this case has the following "request-type" messages: (VI\_hammer used as an example)

Packet --> RubyRequest --> RequestMsg[VI] (L1-->L2) --> RequestMsg (L2-->dir) --> MemoryMsg

So you'll also need to add coreID to RequestMsg (and RequestMsgVI) (gem5-gpu/src/mem/protocol/VI\_hammer-msg.sm)

and in MemoryMsg (gem5/src/mem/protocol/RubySlicc\_Types.sm)

Then for each \_action\_ in the SLICC files which enqueue's a request-type message, copy the corresponding values of interest. I only did this for GET-type requests:

+       out\_msg.coreID := in\_msg.coreID;

Note that in order for "in\_msg" to be defined, the enqueue(...) { } call needs to be wrapped with a "peek" call referencing the input queue and input request type.

Here is a diff for the ruby part of my implementation for clarity: <http://pastebin.com/dyW7isU1> (using MESI\_CMP\_Directory, but still applicable)

--

Matt Poremba

Ph.D. Candidate

354B IST Building

Pennsylvania State University

University Park, PA 16802

- show quoted text -

|  |  |  |
| --- | --- | --- |
|  | **Jintack Lim** | 4/17/14 |

Hi Matt,

This helps me a lot!!

Thank you very much!!

I'll try and will post the result with details so that other guys may refer to in the future.

- show quoted text -

|  |  |  |
| --- | --- | --- |
|  | **Jintack Lim** | 4/19/14 |

Hi Matt and Joel,

I have one missing link now.

Until now,

1. set coreID to each Packet --> Done

2. pass coreID of Packet to coreID of RubyRequest --> Done

3. pass coreID of RubyRequest to coreID of RequestMsg --> Missing Link

4. pass coreID of RequestMsg to coreID of MemoryMsg --> Done

I know this is in your MESI\_CMP\_directory-L1cache.sm,

line# 18,19 of [http://pastebin.com/dyW7isU1](http://www.google.com/url?q=http%3A%2F%2Fpastebin.com%2FdyW7isU1&sa=D&sntz=1&usg=AFQjCNG-0UvWsNQr1H3YailiwRcWsjKuDg)

However, VI\_hammer-CPUCache.sm is somewhat different from your file.

More specifically,

after peek mandatoryQueue\_in to get RubyRequest,

there is no code to copy it to RequestMsg.

(I'm looking at line 408 in VI\_hammer-CPUCache.sm)

In addition, "action(a\_issureGETS..." seems to make RequestMsg, (line 539)

but this does not peek before enqueue, so we can not get RubyRequest.

When I put any coreID here, I can check the value in RubyMemoryController.cc

out\_msg.coreID := 1345;

Could you give some advice?

I always appreciate your help!

Jintack

- show quoted text -

|  |  |  |
| --- | --- | --- |
|  | **Jintack Lim** | 4/19/14 |

Well,

I try to put peek before enqueue, just like this.

539   action(a\_issueGETS, "a", desc="Issue GETS") {

540     peek(mandatoryQueue\_in, RubyRequest) {

541     enqueue(requestNetwork\_out, RequestMsg, latency=issue\_latency) {

542       assert(is\_valid(tbe));

543       out\_msg.Addr := address;

It works for this action.

What I'm not sure is which queue I should "peek" for each action. :)

- show quoted text -

|  |  |  |
| --- | --- | --- |
|  | **Joel Hestness** | 4/24/14 |

**Other recipients:** incredi...@gmail.com

Hi Jintack,

  I'm afraid I can't really give you specific advice, since it seems like these may need to be handled on a case-by-case basis.  I'd recommend being very careful modifying these:

  The potential problem is that the request that is at the head of a particular queue (i.e. visible using peak()) may not be associated with the action that is being triggered.  The a\_issueGETS action is probably okay, because it is only triggered from transitions that are observing the mandatory queue, and the mandatory queue gets popped at the end of the transition, indicating that the head of the queue is the request that is being serviced.  I would think this should work for all the demand misses to the caches, but it doesn't really make sense for writebacks, since the requesting core is not trying to access the data being written back.

  I'm not sure I can give more detailed info without really digging in.  Hopefully, the above will be sufficient.  If you're able to get this to work, it would probably be useful to others if you could give a bit more detail on how you addressed this.

  Thanks!

  Joel

- show quoted text -

2 messages have been deleted.

|  |  |  |
| --- | --- | --- |
|  | **Joel Hestness** | 1/16/16 |

**Other recipients:** 15053...@mail.nwpu.edu.cn

On Fri, Jan 15, 2016 at 12:07 AM, 李晓锋

Hi Joel,

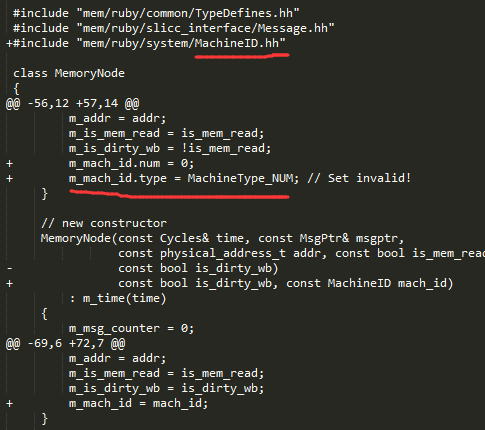
I read your advice in the web(<https://groups.google.com/forum/?hl=zh-cn#!topic/gem5-gpu-dev/25dw-GTHd9E>).

In gem5\_mem\_ctl\_pass\_mach\_id ,MachineID is to distinguish CPU cache, GPU cache, GPU copy DMA, Directory, etc., rather than to distinguish between CPU or GPU. (gem5/src/ruby/commom/MachineID.hh)

But my version is different from the file directory and file content which in you mentioned two patches. I do not know now there is no way to solve this problem? My research is very basic, and the teacher does not give guidance, so I can only go online to check, and do not have a very detailed solution.

If you can give me some help,then I would be very grateful.Thanks!!!

Xiaofeng Li



Hi Li,

  Unfortunately, I have not had a chance to update the required changes to make the latest gem5-gpu pass MachineIDs to the memory controllers from Ruby. As I stated in the thread you link above, this is necessary since the gem5-gpu changes in Feb. last year, because Ruby no longer has direct access to the memory controllers. Because of my inexperience with this aim in the latest codebase, I can only offer general guidance rather than specific instructions for how to make this work.

  Speaking generally, what you will need to do is modify the code in gem5/src/mem/ruby/slicc\_interface/AbstractController.cc to pass the MemoryNode's original requestor MachineID out of the directory controllers and over to the memory controllers. You'll need to modify the queueMemoryRead() and queueMemoryWrite() functions, and maybe even change their function headers so that you can pass the MachineID of the original requestor MachineID to be sent out of the directory. The code for the directories is in src/mem/protocol/VI\_hammer-dir.sm, and you'll probably need to modify near where queueMemory\*() functions are called to get the original requestor. To be more specific than this would be difficult without spending some time with the code.

  Joel

--

  Joel Hestness

On Thu, Jan 14, 2016 at 12:42 AM,

Hi Matt,

I always read your advice and throught ,but I still don't know which files I need to modify ,how to modify. can you give me some help ? Thanks!

Li

在 2014年4月16日星期三 UTC+8下午11:32:34，Matt Poremba写道：

- show quoted text -

--

  Joel Hestness

  PhD Candidate, Computer Architecture

  Dept. of Computer Science, University of Wisconsin - Madison

<http://pages.cs.wisc.edu/~hestness/>

This message has been deleted.

|  |  |  |
| --- | --- | --- |
|  | **iamli...@gmail.com** | Apr 18 |

How do you deal with no "peek" actions finally?

Thanks in advance.

在 2014年4月19日星期六 UTC+8上午1:02:28，Jintack Lim写道：

- show quoted text -

Groups

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 11 of 99+ **(99+)** | | |

****

****

My groups

Home

Starred

Favorites

Click on a group’s star icon to add it to your favorites

Recently viewed

Recent searches

timeout (in littleproxy)

Recently posted to

MV5sim

[Privacy](https://www.google.com/intl/en/policies/privacy/) - [Terms of Service](https://www.google.com/intl/en/policies/terms/)

|  |  |
| --- | --- |
| gem5-gpu Developers List › |  |
|  |  |
| CPU/GPU ID?  13 posts by 4 authors |  |

|  |
| --- |
| 4/12/14  **Jintack Lim** Hi, I'm working on the memory system, RubyMemoryController.cc. For some reason, I need CPU/GPU ID which made memory request. Could you tell me how I can do it? Jintack |

|  |
| --- |
| 4/13/14  **Joel Hestness** Hi Jintack, You would need to modify a few things to pass this information with the memory access: 1) In the packet class (gem5/src/mem/packet.\*), you'll need add fields to track the core type/ID as necessary 2) In the RubyRequest (gem5/src/ |

|  |
| --- |
| 4/16/14  **Jintack Lim** Thanks Joel. Now, I'm at step 4). I can check coreID and coreType at Sequecner.cc. I added coreID and coreType to Packet class. The big picture I have is to convert Packet --> RubyRequest --> MemoryMsg. However, I have trouble to go through cache |

|  |
| --- |
| 4/16/14  **Matt Poremba** Hi Jintack, I did something similar a few years back (for CPU-only gem5/ruby), maybe I can provide more details about the request conversions. The C++ code generated by SLICC allocates a new request at each cache/directory level, so you will need |

|  |
| --- |
| 4/17/14  **Jintack Lim** Hi Matt, This helps me a lot!! Thank you very much!! I'll try and will post the result with details so that other guys may refer to in the future. |

|  |
| --- |
| 4/19/14  **Jintack Lim** Hi Matt and Joel, I have one missing link now. Until now, 1. set coreID to each Packet --> Done 2. pass coreID of Packet to coreID of RubyRequest --> Done 3. pass coreID of RubyRequest to coreID of RequestMsg --> Missing Link 4. pass coreID of |

|  |
| --- |
| 4/19/14  **Jintack Lim** Well, I try to put peek before enqueue, just like this. 539 action(a\_issueGETS, "a", desc="Issue GETS") { 540 peek(mandatoryQueue\_in, RubyRequest) { 541 enqueue(requestNetwork\_out, RequestMsg, latency=issue\_latency) { 542 assert(is\_ |

|  |
| --- |
| 4/24/14  **Joel Hestness** Hi Jintack, I'm afraid I can't really give you specific advice, since it seems like these may need to be handled on a case-by-case basis. I'd recommend being very careful modifying these: The potential problem is that the request that is at |

2 messages have been deleted.

|  |
| --- |
| 1/16/16  **Joel Hestness** On Fri, Jan 15, 2016 at 12:07 AM, 李晓锋 <15053...@mail.nwpu.edu.cn> wrote: > Hi Joel, > > I read your advice in the web( > https://groups.google.com/forum/?hl=zh-cn#!topic/gem5-gpu-dev/25dw-GTHd9E > ). > In gem5\_mem\_ctl\_pass\_mach\_id ,MachineID is to |

This message has been deleted.

|  |  |  |
| --- | --- | --- |
|  | **iamli...@gmail.com** | Apr 18 |

How do you deal with no "peek" actions finally?

Thanks in advance.

在 2014年4月19日星期六 UTC+8上午1:02:28，Jintack Lim写道：

- show quoted text -