Subject: Ptolemy II perfomance
Posted by Sergey Linev on Mon, 26 Apr 2004 08:13:15 GMT
View Forum Message <> Reply to Message

Hello everybody

I want to discuss here some questions about Ptolemy II and its performance.

runs under Linux without any problem. All examples are perfectly working.

Then I created my own actor to test such possibility. This is also not a big problem. I just use several Ptolemy II classes as template (not in sense of C++) and was able to produce actors like network switch or packets buffer. Finally I create scheme for Barrel shift algorithm with 4 buffers and 4 event builders. It works.

Next my step was to test performance of such setup. From my point of view, the main characteristic of performance should be transaction rate or how many data transfers between different entities (actors) are happened during 1 sec of real time (not a simulated time). With my simple setup 4+4 (4 buffers, 4 event builders) I get transaction rate of about 50000 transactions/sec. But when I increase number of nodes to 100+100, transaction rate degrade to about 3000 transactions/sec. My suspicion that decrease was caused by larger buffer depth, needed for Barrel-shift algorithm in case of 100 nodes.

Therefore I modify my actors in the way that I perform only pure data transfer without any buffering and without any modification of tokens (transferring entities in Ptolemy). In the beginning I just create single token and periodically sends it to all buffers. Each buffer immediately forwards it to switch. And switch distributes uniformly all incoming data to outputs without any address analysis.

I test such data transfer network with different number of nodes. Results are:

2+2: ~70000 trans./sec.
10+10: ~60000 trans./sec.
100+100: ~13000 trans./sec.
1000+1000: ~900 trans./sec
10000+10000: out of memory exception

I test it on GSI installation of Debian 3.0 linux, Athlon 1800+, 512 Mb RAM.

My observation was, that 1000+1000 model consumes about 68 Mbytes of memory while 100+100 model requires only 25 Mbytes. Probably, there is a limitation in Java, which is not allows to allocate so much space in memory and starts to use swapping space.

Therefore I want to know from Ptolemy II users, what is a way to specify bigger memory space to Java? Probably, there are other aspects, which should be taken into account.

If it is interesting, I can provide here all my Ptolemy II code and generated xml files.

Subject: Re: Ptolemy II perfomance
Posted by Elena Litvinenko on Mon, 26 Apr 2004 11:36:02 GMT
View Forum Message <> Reply to Message

Hello, everybody,

It must be Java memory restriction (http://java.sun.com/docs/books/jls/first_edition/html/11.doc.html#44395), and it is not clear yet if it is possible to get over. I am also interested to find a decision of such problems often happening in PtolemyII. On many systems 64 Kb is the record size for the Java memory manager. Java completely removes the memory management load from the programmer. As far as I know, there is the discussion of the ways of the Java memory model improvement between Java developers (http://www.theserverside.com/news/thread.tss?thread_id=24199)

Good luck,
Elena Litvinenko

Subject: Re: Ptolemy II perfomance
Posted by Sergey Linev on Mon, 26 Apr 2004 13:00:46 GMT
View Forum Message <> Reply to Message

As was proposed by Ivan Kisel,
I add JAVAFLAGS=-Xmx256m variable, which specify heap size, used by running JAVA program. Seems to be, that is not memory size problem at all. When I look in memory consumption via "top", I see, that my 1000+1000 nodes example consumes about 68 MBytes of memory. Via JAVAFLAGS I specify 400 MBytes of heap and run my 1000+1000 test again. And it again produce only about 900 transactions/sec.

Probably, this is pure Ptolemy II issue, which implements non effective scheduling at all? Or this is problem to operate in JAVA more than 5000 objects simultaneously?

In next few days I will test simplest model with chain of TimedDelay actors.

Subject: Further tests of Ptolemy II Perfomance
Posted by Sergey Linev on Mon, 26 Apr 2004 16:13:17 GMT
View Forum Message <> Reply to Message

I made further tests with Ptolemy II.

I create several models, which are just chain of standard TimedDelay actors. In the beginning I put Clock actor, which generate sequence of 0 and 1 with interval 1 sec. All these tokens just transferred further by each TimedDelay actors with delay of 1 sec. In the end all tokens are

at the end.

There are four files:

File----------actors---tokens---transf----exec---rate--memory

chain_10.xml-------10---100000--~1000000----14s--~70000---15M
chain_100.xml-----100----10000--~1000000----25s--~40000---16M
chain_1000.xml---1000-----1000---~500000----88s---~5000---28M-
chain_10000.xml-10000------100-----~5000-->1000s--~5?----168M

In this table:
actors - number of TimedDelay actors in chain
tokens - number of generated tokens by Clock actor
transf - total number of data transfers during run
exec - execution time on Athlon 1800M+, 512Mb RAM
rate - transfers per second
memory - used memory size as shown by top

First two files can be viewed and run with vergil, with other can be some problems.

I run these examples with ptexecute routine like:
> $PTII/bin/ptexecute chain_100.xml

Variable JAVAFLAGS was set to use 400 Mbytes memory for heap.
JAVAFLAGS=-Xmx400m

How I can explain and improve these results?

File Attachments

1) chain_10.xml, downloaded 1166 times
2) chain_100.xml, downloaded 1143 times
3) chain_1000.xml, downloaded 1211 times

Subject: SystemC versus Ptolemy II perfomance
Posted by Sergey Linev on Wed, 28 Apr 2004 09:17:39 GMT
View Forum Message <> Reply to Message

To be able compare Ptolemy and SystemC, I write a small program using SystemC classes.

I create TToken class, which similar meaning as in Ptolemy. This should be abstract class for data containers, which should be transferred between model components (actors).

To perform data exchange between components, three classes were introduced:

In addition, I create TActor class, which can be used as base class for all model components.

To configure model, comparable with Ptolemy model, which I use in performance tests, I introduce three classes:

From these components I construct very similar simulation, as I did before with Ptolemy. I configure chain of components, which start with TGenerator and stops by TDiscard modules, and in between arbitrary number of TTimedDelay actors can be inserted.

To compile this program, first SystemC 2.0.1 should be installed. Then test.tar.gz archive should be unpacked. In Makefile correct path to SystemC installation should be specify. Then

modified to be used in Linux, but I see no problem to compile examples under another platforms. I am not sure, if SystemC works under cygwin, but it should works with MS VC compiler.

I run my tests on the same machine (Athlon 1800M+, 512 MB RAM, Debain 3.0, gcc 2.95.4), where I run Ptolemy tests. Results are:

```
> run.x 10 1000000

Create 10 delay actors
Execute done in 12 sec
Number of transactions = 10999955
 Rate = 916663  trans/sec

> run.x 100 100000
Create 100 delay actors
Execute done in 14 sec
Number of transactions = 10095181
 Rate = 721084  trans/sec

> run.x 1000 10000
Create 1000 delay actors
Execute done in 36 sec
Number of transactions = 9527192
 Rate = 264644  trans/sec

> run.x 10000 10000
Create 10000 delay actors
Execute done in 283 sec
Number of transactions = 51726725
 Rate = 182780  trans/sec
```

Memory usage (as reported by top):

Can be seen, that transaction rate in SystemC about 10 time faster than in Ptolemy II in case

of small number of components (less than 100). If one use bigger number of actors in SystemC, it just introduce factor of 5 penalty, while in Ptolemy it is two or three orders of magnitude.

On my machine test with 10000 node Ptolemy finish in 6 hours and makes only 5000 transactions, which means 0.25 trans/sec speed.

Any comments?

File Attachments

1) test.tar.gz, downloaded 1306 times

Subject: Re: SystemC versus Ptolemy II perfomance
Posted by Ivan Kisel on Wed, 28 Apr 2004 14:33:08 GMT
View Forum Message <> Reply to Message

Dear Sergei,

I don't expect that Ptolemy will be faster than SystemC.

My statement is that Ptolemy with his graphical interface vergil is more suitable for developing, debugging and investigating the design. SystemC has no graphical interface. This will result in your/my/our time, not CPU, when you start creating large complicated systems, but not simply increasing number of actors.

Ivan.

Subject: Re: SystemC versus Ptolemy II perfomance
Posted by Sergey Linev on Wed, 28 Apr 2004 17:44:04 GMT
View Forum Message <> Reply to Message

Dear Ivan

I never say, that we must use SystemC somehow.

I only want to stress, that Ptolemy has strong limitation, then number of components exceed 100-200 actors and cannot really works at all, when number of actors more than 1000. SystemC has no such strong limitation.

Probably, there is a solution for Ptolemy, because results looks very stupid. I already ask Ptolemy developers, but get no response till now.

Subject: Re: SystemC versus Ptolemy II perfomance
Posted by Walter F.J. Müller on Wed, 28 Apr 2004 18:01:18 GMT
View Forum Message <> Reply to Message

S. Linev wrote on Wed, 28 April 2004 19:44...I already ask Ptolemy developers, but get no response till now...

I guess there is a forum or at least mailing list archive for Ptolemy. If there is, you probably searched for "performance" or some other buzz words. Looking into such a forum/archive gives a picture on what 'typical' Ptolemy users do and what their problems are. So I wonder, whether the system sizes you see the performance degrade for are average or large compared with what other users usually do.

Subject: Re: SystemC versus Ptolemy II perfomance
Posted by Sergey Linev on Thu, 29 Apr 2004 06:37:58 GMT
View Forum Message <> Reply to Message

I did exactly this.

There is comp.soft-sys.ptolemy usenet group, which is now situated in google:

http://groups.google.com/groups?group=comp.soft-sys.ptolemy

I serach for perfomance and speed question in past several years, but did not find any relevant topic for this.

Another place is
http://ptolemy.eecs.berkeley.edu/search/

There is message, but it is 8 years old.
http://ptolemy.eecs.berkeley.edu/mailing-lists/split/9607/Re:_Ptolemy_is _slow_60

Subject: Re: Ptolemy II perfomance
Posted by Walter F.J. Müller on Fri, 07 May 2004 19:45:32 GMT
View Forum Message <> Reply to Message

Some results of a simple benchmark test were presented in the CBM DAQ/Trigger meeting and are accessible under http://www.gsi.de/documents/DOC-2004-May-82-2.pdf .

The reasons for the strong performance drop with Ptolemy for larger systems is under investigation, but any tip is welcome.

Subject: Re: Ptolemy II perfomance
Posted by Krzysztof Korcyl on Thu, 13 May 2004 12:29:30 GMT
View Forum Message <> Reply to Message

Hello,

To somehow conclude on the modeling environment's performance studies we would like to

repeat the same tests but with the Ptolemy Classic.
Would it be possible to get sources of objects (modules) used in the SystemC, which seems to be close to C++, hence also to the Ptolemy Classic. We could then translate behaviour into Ptolemy Classic rules and make corresponding measurements.

best regards,

kk

Subject: Re: Ptolemy II perfomance
Posted by Sergey Linev on Thu, 13 May 2004 16:14:27 GMT
View Forum Message <> Reply to Message

I include all my code, that I use in tests, in my previous messages.

I think, that test can be done with native Ptolemy Classic classes. When I test Ptolemy II, I just took Clock, TimedDelay and Discard actors. Very similar actors should exists in Ptolemy Classic too.

If no, any kind of simple model with a lot of similar components is suitable. Idea of all my tests is to measure how many transactions between model entities happens per second and does this value scales with number of entities.

Subject: Re: Ptolemy II perfomance
Posted by Krzysztof Korcyl on Fri, 14 May 2004 07:45:13 GMT
View Forum Message <> Reply to Message

Helo Sergey,

> I include all my code, that I use in tests, in my previous messages.

Unfortunately, your attachment has been filtered out by the virus protection on some mailer system. Details of that action are attached at the end of this message.
Could you please put your code on some web site where we could have access to and fetch it later?

cheers,

kk.

****

This is a message from the MailScanner E-Mail Virus Protection Service
---
The original e-mail message contained potentially dangerous content,
which has been removed for your safety.

The content is dangerous as it is often used to spread viruses or to gain personal or confidential information from you, such as passwords or credit card numbers.

Due to limitations placed on us by the Regulation of Investigatory Powers Act 2000, we were unable to keep a copy of the original attachment.

The content filters found this:
  Found a script in HTML message

--
Postmaster

MailScanner thanks transtec Computers for their support

Subject: Re: Ptolemy II perfomance
Posted by Sergey Linev on Fri, 14 May 2004 08:09:43 GMT
View Forum Message <> Reply to Message

Hallo, Krzysztof

Attachments are in my messages, that I put in the forum. I do not think, that they are automatically distributed over the mail notification system.

Anyway, my SystemC code now also avaliable at:

http://www-linux.gsi.de/~linev/systemctest.tar.gz

Hope, you can download it.

Subject: Re: Ptolemy II perfomance
Posted by Elena Litvinenko on Fri, 14 May 2004 09:42:22 GMT
View Forum Message <> Reply to Message

Hello,

Sergey, I slightly modified your SystemC example code to force it work under Windows (and MS VC 6.0 with SP5). I removed protection for the methods of the classes TActor, TDiscard, and TTimedDelay, and did some minor changes in the MainProgram.cpp. The results was executed under WinXP (Pentium 4, 2.53GHz, 512Mb RAM) and results look like

---
Nodes Memory Transact Time Rate
10 1.2M ~1x10^5 1s ~1.09 x 10^5
100 1.9M ~1x10^6 7s ~1.43 x 10^5

1000 9.4M ~1x10^7 91s ~1.04 x 10^5
10000 84.1M ~5x10^7 595s ~0.86 x 10^5

The chanded code, dsp and exe files are in the attachment.

Regards,
Elena Litvinenko

File Attachments
1) systemc_test_win.zip, downloaded 1301 times

Subject: Re: Ptolemy II perfomance
Posted by Sergey Linev on Fri, 14 May 2004 10:46:24 GMT
View Forum Message <> Reply to Message

Hello, Elena

Except first test with 10 nodes, results on Linux and Windows looks similar. In general, there are no reasons, why SystemC simulation can not work under Windows as long as it is supported SystemC platform.

Subject: Re: Ptolemy II perfomance
Posted by Walter F.J. Müller on Fri, 14 May 2004 10:52:22 GMT
View Forum Message <> Reply to Message

Krzysztof Korcyl wrote on Fri, 14 May 2004 09:45...
Could you please put your code on some web site where we could have access to and fetch it later?

This forum supports attachments, so one can add files to a posting. Sergey used this mechanism to add the tarball with the test codes to a posting earlier in this topic. Look at the end of

http://forum.gsi.de/index.php?t=tree&th=155&mid=385

and you'll find "Attachment: test.tar.gz". You can download this file directly from the forum. There is no need for putting putting it on the web, its already there...

Subject: Re: Ptolemy II perfomance

Posted by Elena Litvinenko on Fri, 14 May 2004 12:05:36 GMT
View Forum Message <> Reply to Message

S.Linev wrote
"... there are no reasons, why SystemC simulation can not work under Windows as long as it is supported SystemC platform"

If you meant code modifications, it was done, because otherwise compilation failed.

Subject: Re: Ptolemy II perfomance
Posted by Sergey Linev on Fri, 14 May 2004 12:51:11 GMT
View Forum Message <> Reply to Message

Unfortunately, I never test SystemC under Windows.
Probably, one should follow defined common programming style to avoid problems between Windows and Linux, if both platform should be used.

Till now no clear understanding at all, which platform - Ptolemy II, Ptolemy Classic or SystemC 2.0 should be used. I am not in favour of any of them as long as I use non of them in real application. Probably, C++ preferable for reasons like integration with other platforms.

Subject: Re: Ptolemy II perfomance
Posted by Krzysztof Korcyl on Tue, 25 May 2004 10:18:32 GMT
View Forum Message <> Reply to Message

Dear Sergey,

I need a bit more clarifications on the test with systemC with chain of delays. I do not know the systemC but from the MainProgram.cpp I am guessing the following (big capital letters are comments to the MainProgram.cpp code):

A. system is composed of one generator,

```
TGenerator* generator = new TGenerator("Generator", 1.);
```

B. there is a variable number of delays, where each consecutive delay produces shorter delay (range from 0.999 - 0.899). The delay object gets a message on it's input and relays it to it's output after it's internal delay.

```
TTimedDelay* delays[numdelays];
for (int n=0;n<numdelays;n++)
  delays[n] = new TTimedDelay(mname("D",n), 0.999 - 0.1*n/numdelays);
```

C. There is one sink in the system which deletes all messages arriving on it's input

```
TDiscard* discard = new TDiscard("Discard");
```

D. The delay objects are connected via another objects: token_channel. Each token_channel receives a message on it's input and relays it on it's output after a fixed delay - which is

apprently 0.0 (ie the token_channel is infinitively fast). I understand, that this is systemC requirement to use such token_channels to connect objects (in general one can assign a non zero delay to the token_channel). It that correct?

```
TToken_channel* chanels[numdelays+1];
for (int n=0;n<=numdelays;n++)
  chanels[n] = new TToken_channel(mname("C",n), 0.);
```

E. Below is code making connections between delay objects and token_channels

```
generator->output(*chanels[0]);
for (int n=0;n<numdelays;n++) {
  delays[n]->input(*chanels[n]);
  delays[n]->output(*chanels[n+1]);
}
```

F. Here you end the chain with discard object.

```
discard->input(*chanels[numdelays]);
```

The simulation operates as follows:
Generator produces a message and passes it into the first token_channel. The first delay gets the message and after it's delay puts it on output. The second token_channel gets the message and passes it immediately to the next delay object and so on. The minimum inverval between two consecutive messages from the generator is greater (1.0) than maximum delay (0.999), thus we will never have problem of buffering message due to some channel being occupied by previous message.

What was your measure when running the simulation: number of messages generated by the Generator?

cheers,

Krzysztof

Subject: Re: Ptolemy II perfomance
Posted by Sergey Linev on Tue, 25 May 2004 10:47:46 GMT
View Forum Message <> Reply to Message

Dear Krzysztof

You are very precisely describe algorithm, how my simple model is working. I use decreasing of time interval to avoid buffering inside TimedDelay actors.

In my view, this is main difference of Ptolemy and SystemC. Ptolemy guarantee sequnce, in which actors will be fired (activated) when several messages (tokens) have similar time stamp. Contrary to Ptolemy in SystemC any message with same time stamp can be processed first. Therefore, in Ptolemy model, when I have chain of TimedDelay actors with the same delay is working perfectly, while in SystemC such model frequently lost data, if you do not place buffer inside.

Channels are not explicitly required in SystemC. Generally, one can use standard methods to connect input and output ports. I introduce my own channel just to have flexibility to put some functionality in it like delays, paket loss, data corruption and so on.

In all my measurements I just count number of transactions over token channels and estimate its rate over physical time. For instance, my simple model with 10 delay actors makes output:

Create 10 delay actors
Execute done in 12 sec
Number of transactions = 10999955
Rate = 916663 trans/sec

This means, that during 12 second of physical computer time I perform 10999955 transactions over all channel, which are created in model. To count them, I just increment static member in my TokenChannel class.

Subject: Re: Ptolemy II perfomance
Posted by Krzysztof Korcyl on Tue, 25 May 2004 16:32:26 GMT
View Forum Message <> Reply to Message

Hello Sergey,

When you refer to the execution time - does it include time spent on creation of all the objects (actors)? - or the time measurement is started after the whole chain is constructed and the messages start to flow?

cheers,

Krzysztof

Subject: Re: Ptolemy II perfomance
Posted by Sergey Linev on Wed, 26 May 2004 06:56:49 GMT
View Forum Message <> Reply to Message

Hello Krzysztof

I measure time only for model execution. Creation time of all actors in SystemC is negligible small. With Ptolemy my observation is that for big (>1000 actors) models creation takes several minutes.

Subject: Re: Ptolemy II perfomance
Posted by Krzysztof Korcyl on Tue, 08 Jun 2004 09:35:40 GMT
View Forum Message <> Reply to Message

Dear all,

We completed simple performance study with the Ptolemy Classic.
In our tests we used the same model of chained delays as it was tested with the SystemC and Ptolemy II.
We experimented with two setups. The first one was a chain of linked delay Elements.
In the other setup we interconnected delay Elements via additional objects representing "connection". The latter setup was tried to have the same number of objects as in the SystemC model.
Diagrams of both setups are pesented in the first slides of the presentation attached to this message. The last two slides from the presentation show comparison between the three modeling environments (Ptolemy II, SystemC and Ptolemy Classic). The last two environments, based on C++, are superior over the Java implementation.

cheers,

Krzysztof.

File Attachments

1) PtolemyPerformance1.pdf, downloaded 1847 times