

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 1058 times
  - 2) [chain\\_100.xml](#), downloaded 1037 times
  - 3) [chain\\_1000.xml](#), downloaded 1093 times
-