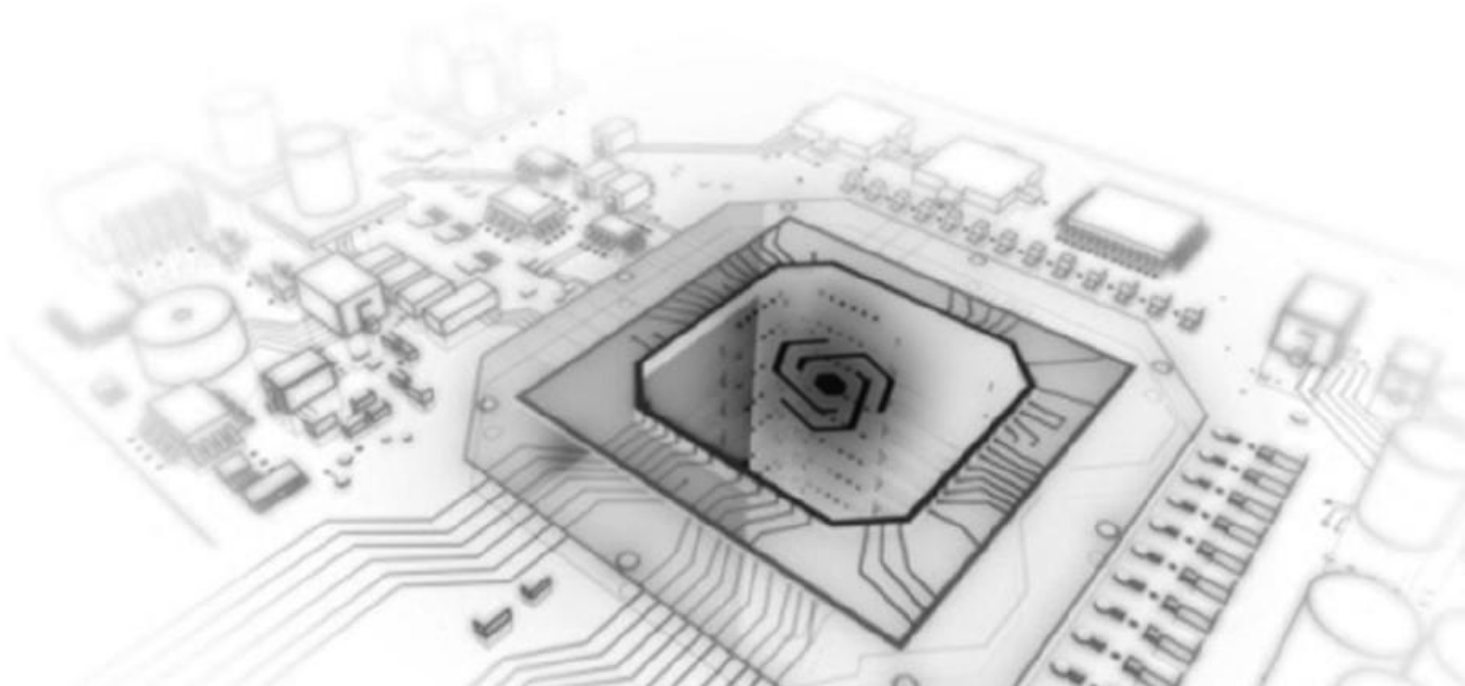


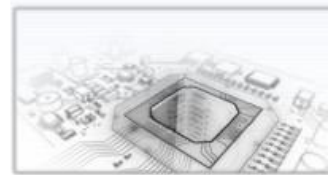
Паралелно Програмиране

Паралелни компютърни симулации в средата на Omnet++ с MPI



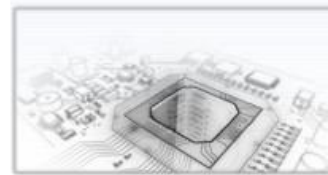
Съдържание

1. Основни характеристики на Omnet++ симулатора
2. Паралелни симулации
3. Подсистема за паралелни симулации при Omnet++
4. Демонстрация на възможностите за паралелни симулации при Omnet++ симулатора



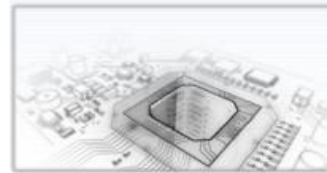
Основни характеристики на Omnet++ (1)

- OMNeT++ представлява разширяема, модулна, C++ базирана библиотека и работна рамка за симулации която включва интегрирана графична среда за развитие.
- Основното приложение на OMNeT++ е в областта на комуникационните мрежи и разпределените системи, но поради гъвкавата архитектура тази среда може да се използва за симулиране на широк спектър комплексни системи и хардуерни архитектури.
- Много често OMNeT++ се нарича мрежов симулатор, но това не е напълно вярно. Симулатора предоставя основни инструменти за описание на модели, докато за специфични симулации в областта на мрежовите и системните архитектури използват се допълнителни рамки. Това позволява OMNeT++ да бъде обща среда за симулиране на най-различни дискретни процеси.



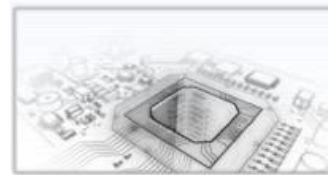
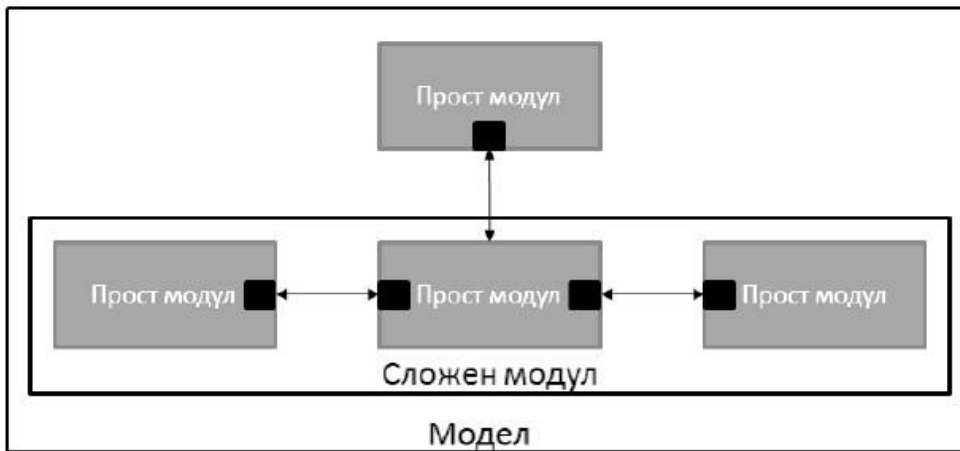
Основни характеристики на Omnet++ (2)

- Основата на OMNeT++ е компонентно ориентирания подход, който насърчава използването на структурирани модели за многократна употреба.
- OMNeT++ използва обща компонент-базирана архитектура, която дава възможност на проектанта на модела да изгради съответствие между концепциите от една страна, и компонентите на модела, от друга страна. Основните компоненти на модела се наричат модули. Модулите могат да бъдат комбинирани по различни начини и да се използват повечекратно. Повече прости модули могат да бъдат капсулирани в рамките на сложни модули като нивата на йерархията не са ограничени.
- Комуникацията между модулите се осъществява посредством обмен на съобщения. Те се обменят през входно/изходни интерфейси наречени портове.



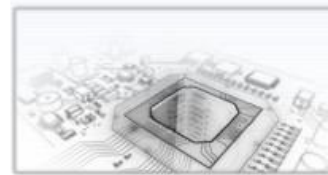
Основни характеристики на Omnet++ (3)

- Симулатора използва собствен DSL (Domain Specific Language) език за дефиниране на елементите на симулацията. Типични елементи на описанието с NED езика са декларации на простите модули, дефиниции на сложните модули и дефиниции на мрежата. Една от главните характеристики на този език е възможността за повечекратно използване на един и същ модул с различни параметри.
- В общи линии, активните елементи в OMNeT++ се имплементират на C++ с използване на клас библиотека за симулации, докато декларациите на модулите и мрежата се дефинират в NED езика.



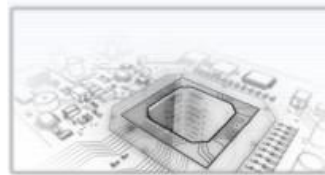
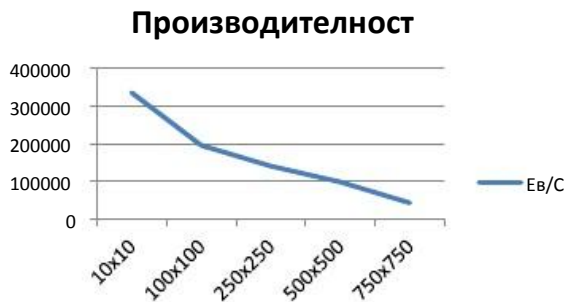
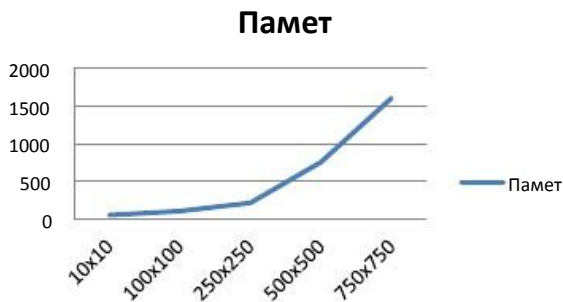
Паралелни симулации (1) (1)

- В следствие на бързото развитие на системите архитектури и тенденцията за изграждане на комплексни паралелни компютърни платформи предвиждането на производителността на системите и елиминацията на възможните недостатъци става все по-сложна.
- Проектирането на съвременните компютърни системи с висока сложност добавя ново измерение към архитектурния синтез. Единият от основните проблеми е да се разгледат взаимоотношенията помежду основните компоненти на системата или свързващата мрежа.
- Един от най-подходящите методи за верификация на поведението и производителността на системните архитектури и мрежи е компютърната симулация. Компютърните симулации могат да бъдат разделени в повече групи: симулации на дискретни събития, продължителни симулации, Монте Карло симулации и много други.



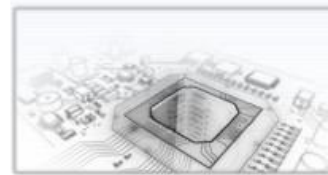
Паралелни симулации (2) (2)

- Симулаторите на дискретни събития изискват големи изчислителни ресурси. Главна причина за това е високата сложност на съвременните архитектури и мрежи които са симулират.
- Много често симулациония модел надхвърля ограничението на паметта при единна работна станция.
- На следващата фигура представени са резултати от последователно изпълнение на симулация в зависимост от сложността на модела.



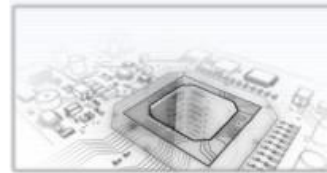
Паралелни симулации (3) (3)

- Паралелната симулация на дискретни събития (PDES - Parallel discrete event simulation) предлага атрактивна алтернатива. Посредством разпределение на симулацията на повече процесори, може да се съкрати времето за изпълнение в отношение на последователната симулация и едновременно да се изпълнят високите изисквания към размера на паметта. Следователно PDES предоставя възможност за детайлно симулиране на много сложни системи, като съкращава времето за изпълнение.

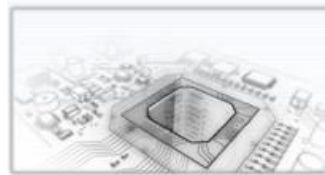
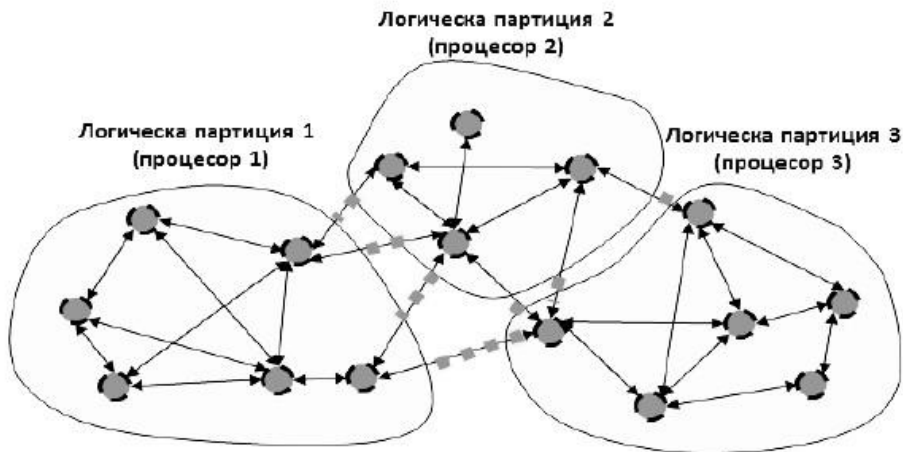


Подсистема за паралелни симулации при Omnet++ (1)

- OMNeT++ поддържа механизъм за паралелни симулации с имплементиране на консервативни протоколи за синхронизация.
- Подсистемата за паралелни симулации представлява допълнителна компонента, която може да бъде изключена доколкото не е потребна.
- Главната цел OMNeT++ е да предостави подход за паралелно изпълнение на сложни симулации, които не изисква модификации на модела.
- При този симулатор не се потребни особени процедури за промяна на сорс кода и топологията. Те са едни и същи и непроменени независимо дали симулацията се изпълнява последователно или паралелно.

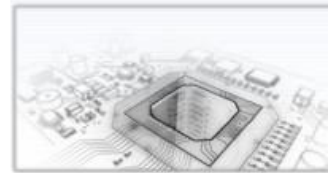


- Целия процес на паралелната симулация е прозрачен за програмиста. Модулите, връзките и останалите елементи на модела се разделят на предварително определени дялове, които се наричат партиции. Всяка партиция се свързва към различен логически процес, като постоянно се поддържа синхронизация и се запазва последователността на събитията.
- По принцип всеки логически процес се изпълнява на различна физическа система, което позволява да се ускори процеса на изпълнение и да се преодолее ограничението на паметта



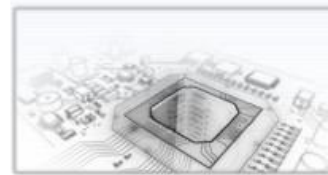
Подсистема за паралелни симулации при Omnet++ (3)

- Почти всеки модел може да се изпълнява на паралелна система, като се вземат в предвид следващите изключения:
- Модулите могат да комуникират само с изпращане на съобщения;
- Забранено е използването на общи променливи;
- Съществуват малък брой на лимитации при директното изпращане на съобщения;
- Задължително е използването на закъснения по каналите ;
- В момента OMNeT++ поддържа паралелно изпълнение само на статични топологии.



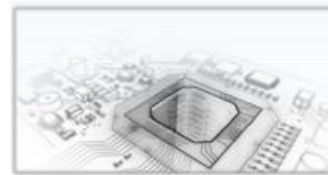
Подсистема за паралелни симулации при Omnet++ (4)

- Тази гъвкавост при изпълнение на паралелни симулации се дължи на модулната архитектура на подсистема за паралелни симулации.
- Самата подсистема се състои от три нива: комуникационно ниво, ниво за разпределение и ниво за синхронизация.



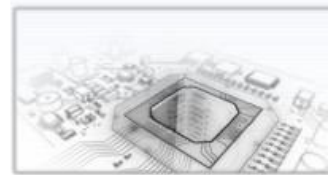
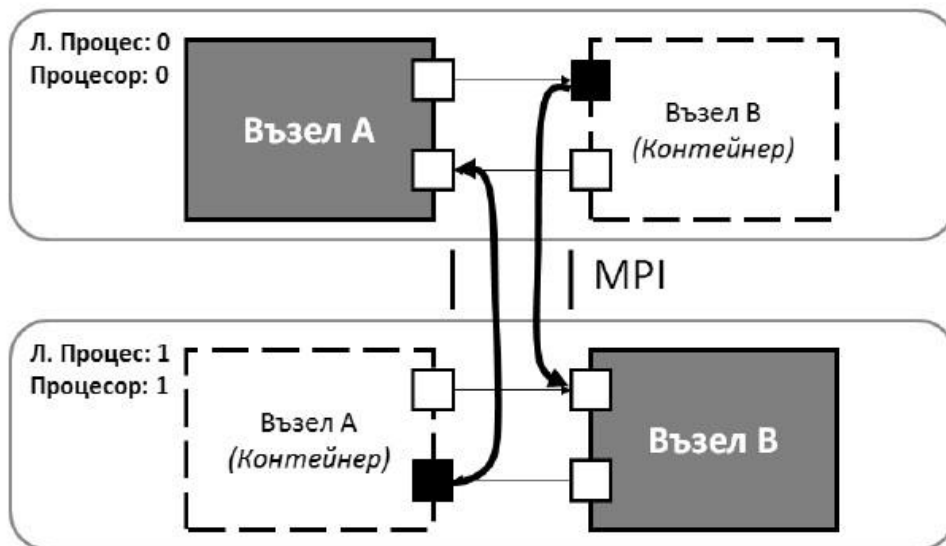
Подсистема за паралелни симулации при Omnet++ (5)

- Главна характеристика на OMNeT++ е възможността за изпълнение на симулации на разпределени системи без да се налага промяна на сорс кода и общия модел.
- Модела на симулацията се разпределя на повече логически процеси, които се изпълняват на различни процесори.
- Особеността която позволява лесно и прозрачно изпълнение на паралелните симулации е използването на виртуални модули наречени контейнери.
- Във всеки локален логически процес, освен разпределените модули, намира се и контейнер в който виртуално са представени подобните модули които се изпълняват на други логически процеси.
- Модулите които се изпълняват на един логически процес могат да изпращат директни съобщения до подобните външни модули използвайки контейнера.
- След това контейнера с използване на различни методи за комуникация(в случая MPI) и прокси портове прозрачно препраща съобщението до „реалния“ модул който се изпълнява на друг логически процес.



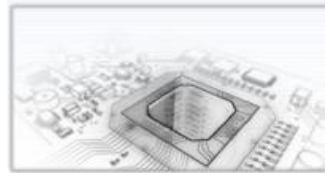
Подсистема за паралелни симулации при Omnet++ (6)

- След това контейнера с използване на различни методи за комуникация(в случая MPI) и прокси портове прозрачно препраща съобщението до „реалния“ модул който се изпълнява на друг логически процес.



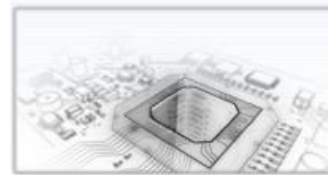
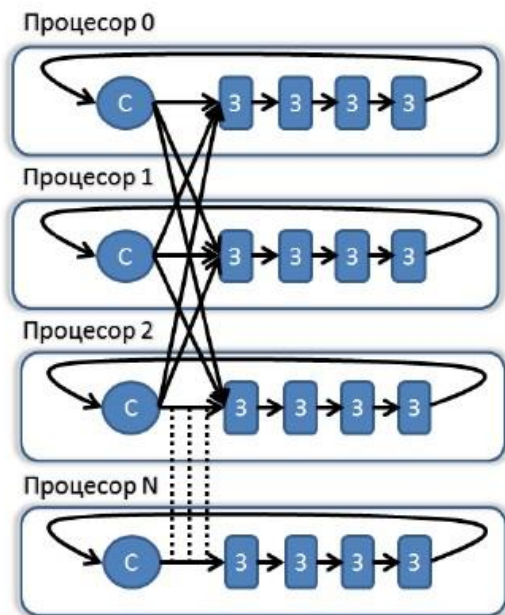
Демонстрация на паралелни симулации (1) (1)

- За демонстриране на възможностите за паралелни симулации в OMNeT++ средата ще използваме модел на затворена мрежа от опашки (Closed Queuing Network).
- Модела е развит от разработчиците на средата и е съставен от N тандеми опашки, като всеки тандем се състои от суич и K сървърни заявки с експоненциално време за обслужване. Последната заявка в тандема се извива назад към суича. Всеки суич по случаен избор разпределя първата заявка към останалите тандеми. При разпределението прави се равномерна дистрибуция на заявките.



Демонстрация на паралелни симулации (2) (2)

- За паралелно изпълнение на този модел, първоначално трябва да се зададат основните конфигурационни параметри. Тези параметри са дефинирани в текстовия файл `omnetpp.ini`.
- В процеса на конфигуриране първата стъпка е да се направи разпределение на модулите помежду процесорите т.е. помежду логическите процеси



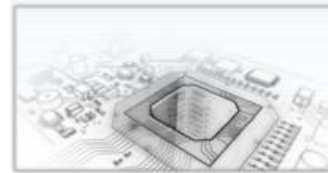
Демонстрация на паралелни симулации (4) и (4)

- Разпределението се прави с добавяне на следващия код към конфигурационния файл:

```
[partitioning]
*.tandemQueue[0]**.partition-id = 0
*.tandemQueue[1]**.partition-id = 1
*.tandemQueue[2]**.partition-id = 2
.....
*.tandemQueue[n]**.partition-id = n
```

- Следващата стъпка е избирането на библиотека за комуникация.

```
[general]
#parsim-communications-class="cFileCommunications"
#parsim-communications-class = "cNamedPipeCommunications"
parsim-communications-class="cMPICommunications"
```



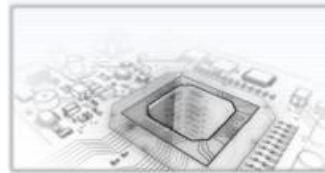
Демонстрация на паралелни симулации (5)

- Също така в конфигурационния файл трябва да активираме подсистемата за паралелни симулации. Това се прави по следващия начин:

```
[general]  
parallel-simulation = true  
#parallel-simulation=false
```

- Както и алгоритмът за синхронизация

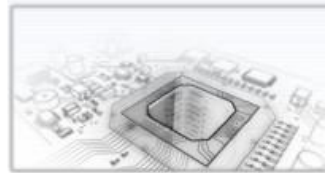
```
[general]  
#parsim-synchronization-class= "cNoSynchronization"  
parsim-synchronization-class = "cNullMessageProtocol"  
#parsim-synchronization-class= "cISPEventLogger"  
#parsim-synchronization-class= "cIdealSimulationProtocol"
```



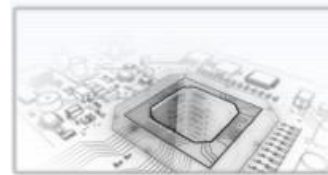
Демонстрация на паралелни симулации (6)

- За изпълнение на паралелната симулацията с MPI използват се стандартните команди. Примерно за стартиране на симулацията на затворената мрежа от опашки на 3 процесора ще използваме следващата команда:

```
mpirun -n3 -machinefile file ./cqn
```



- <http://www.omnetpp.org/documentation>
- “PARALLEL SIMULATION MADE EASY WITH OMNeT++”, A.Varga, Y. Ahmed, G. Egan, European Simulation Symposium (ESS2003), Oct. 2003, Delft, The Netherlands
- “Modeling and Tools for Network Simulations”, K. Wehrle, M. Gunes, J. Gross, Springer, 2010.



Благодаря за вниманието!

