Операционна среда за разпределена симулация

Христо Вълчанов

RunTime System for Distributed Simulation: *Parallel discrete event simulation (PDES) is a widely used approach for investigation and evaluation of complex systems. This paper presents an operational model of a distributed simulation system running on a network of workstations. The system has cluster architecture and uses TimeWarp synchronization protocol between clusters. Issues related to the cost of synchronization are discussed and a comparative evaluation with other operating models is made.*

Key words: *PDES, TimeWarp, Distributed Simulation, RunTime System.*

ВЪВЕДЕНИЕ

Разпределената дискретна симулация, базирана на събития (РДСС) е технология, която позволява симулационната програма да бъде изпълнена върху паралелна/разпределена платформа, постигайки повишаване на производителността на симулацията. При РДСС реалната система се представя като съвкупност от отделни компоненти, наречени *симулационни обекти* (СО). Всеки СО може да се изпълнява върху отделен процесор. Настъпващите в дискретни моменти от моделното време събития се симулират посредством обмен на съобщения между СО, всяко от които има времеви маркер. Поради разпределения характер на симулацията, нейната коректност се постига чрез специални синхронизиращи протоколи. Един от най-използваните протоколи е TimeWarp [1, 2]. Съгласно семантиката на протокола, СО обработват събитията по реда на тяхното пристигане. При всяко събитие локалното моделно време на СО става равно на времевия маркер на събитието. Поради закъсненията в комуникационната среда е възможно получаване на събитие с времеви маркер, по-малък от локалното време на СО. Такова съобщение е известно като *съобщение-нарушител*. За възстановяване на коректността на симулацията СО трябва да върне съхраненото си състояние преди получаването на този нарушител. Същевременно е необходимо да се отменят всички изпратени до този момент съобщения към други СО. Това се реализира посредством изпращане на съответни т.н. *анти-съобщения*. Процесът на възстановяване на състоянието на СО е известен като *връщане назад (rollback)*.

Представянето на множество моделирани компоненти в рамките на отделен симулационен обект може да увеличи грануларността на симулацията [5], позволявайки вътрешната комуникация в симулационния модел да бъде реализирана посредством механизмите на достъп до обща памет. Формираният по този начин клъстер от симулационни обекти (симулационен клъстер) се реализира като самостоятелен процес, изпълняващ се върху отделен изчислителен възел. Групирането на симулационни обекти позволява паралелизмът в симулационния модел да се извършва на ниво клъстер. По този начин отпада както необходимостта отделният СО да има собствен списък на събитията, така и да реализира самостоятелно алгоритъма на синхронизиращия протокол. Вместо това, симулационният клъстер (СК) поддържа общ клъстерен списък на подредени по времената им на настъпване събития (КСС), както и клъстерно виртуално време (КВВ). При обработката на събитие КВВ нараства с времето на неговото настъпване. Така, в рамките на СК, събитията се обработват посредством последователна симулация, чрез което значително се редуцират разходите за синхронизация между СО в клъстера. Между изпълняващите се на отделни изчислителни възли СК се прилага оптимистичния протокол TimeWarp.

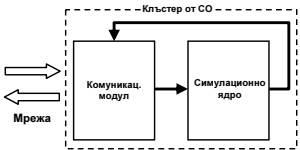
Един от факторите, допринасящи за нарастване на популярността на локалните компютърни мрежи като платформи за изпълнение на разпределени симулационни приложения е наличието на стандартизирани комуникационни пакети и средства. Един от най-често използваните съвременни комуникационни стандарти е

стандартът MPI [4], който дефинира програмен интерфейс и семантика на обмен на съобщения.

В настоящата статия е представена операционна изпълнителна среда за разпределена симулация, функционираща върху мрежа от Linux работни станции и използваща за комуникации MPI. Разгледани са въпроси, свързани с разходите по синхронизация и комуникация. Представени са резултати от сравнителен анализ с други модели на известни системи за разпределена симулация.

ОПЕРАЦИОННИ МОДЕЛИ НА СИСТЕМИ ЗА РДСС

Редица съвременни системи за симулация са реализирани на базата на концептуалния операционен модел WARPED (фиг.1) [3, 6]. В този модел всеки СК се състои от два основни компонента: симулационно ядро и комуникационен модул, реализирани като паралелно изпълняващи се нишки.



Фиг. 1. Операционен модел WARPED

Симулационното ядро (СЯ) обработва подредените по времената им на настъпване в КСС събития. СЯ изпълнява две основни функции- синхронизиране на събитията и изпълнение на кода на СО, моделиращ настъпването на тези събития. Комуникационният модул (КМ) поддържа комуникацията с други СК. Той осигурява както получаването на съобщения за настъпващи събития от други СК, така и предаването на очакващите в изходната опашка съобщения.

Характерно за този операционен модел е, че операциите по синхронизиране се извършват в кода на симулационното ядро, при това действията по синхронизиране на съобщенията и тяхната обработка са последователни операции. СЯ проверява първо за наличие на съобщение във входния буфер и при откриване на такова, стартира алгоритъма за синхронизация. При *съобщение-нарушител*, причинно-следствената грешка се отстранява чрез стартиране на действия по възстановяване на предишното съхранено коректно състояние. Всяко коректно по отношение на моделното време съобщение се поставя в КСС. След тези действия СЯ продължава с обработката на подредените по времената им на настъпване събития в КСС.

Директното прилагане на този операционен модел за разпределена оптимистична симулация върху изпълнителни архитектури с разпределена памет може да доведе до неблагоприятни ефекти, рефлектиращи в намаляване на производителността на симулацията. Характерно за тези архитектури е, че поради редица фактори като: паралелното функциониране на СО, асинхронния характер на обмен на съобщенията между тях и закъсненията, типични за комуникационната среда, съществува висока вероятност за получаване на *съобщение-нарушител*. От друга страна, паралелното функциониране на комуникационния модул и симулационното ядро предполага получаването на съобщенията да бъде независимо реализирано от синхронизацията и обработката на събитията в клъстера.

Обработката на предназначено за СО събитие се изпълнява като атомарна операция, като проверките за пристигнали съобщения-нарушители или анти-съобщения се извършват единствено между две последователни обработки на събития. Това означава, че действията по връщане на състоянието назад, ако са необходими, не могат да прекъснат текущо обработваното събитие.

Вследствие на това могат да се направят изводи за евентуални негативни последици по отношение на производителността на симулацията:

- Многократно нарастване на броя на анти-съобщенията;
- Увеличаване на разходите за връщане назад;
- Повтаряща се обработка и изпращане на съобщения.

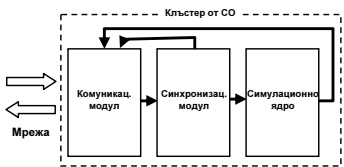
Възможно решение за редуциране на споменатите разходи е интегрирането на функциите на комуникационния и симулационния модули в единичен компонент. Този подход е използван при реализацията на симулаторите GTW и ROSS [6]. При този модел в началото на всеки симулационен цикъл се проверява за наличие на съобщение от комуникационната система, изпълняват се действията по синхронизация, обработва се поредното събитие от списъка на събитията и в края на цикъла се изпращат генерираните по време на обработката съобщения. Моделът е оптимизиран основно за компютърни системи с обща памет.

При архитектури с разпределена памет, каквито са компютърните мрежи, комуникацията се извършва посредством библиотеки за предаване на съобщения. Семантиката на библиотеките по стандарта MPI определя всяка *send* операция да бъде завършена единствено при изпълнението на съответната ѝ *receive* операция. Благодарение на тази особеност се ограничава възможността симулационните обекти да изпращат повече съобщения, отколкото техните получатели могат да обработят. Като резултат е налице по-рядко настъпване на ситуации по връщане назад, което би рефлектирало в по-добри времеви параметри на симулацията.

Това предимство на MPI обаче се явява съществен недостатък в конкретния случай. Семантиката на симулацията на динамични дискретни системи изисква блокиране на обекта-създател до момента на стартиране на новосъздадения обект. При създаване на СО в друг клъстер, обектът-създател остава в блокирано състояние. Потвърждението за създаването на новия обект не може да бъде получено, тъй като това се извършва в началото на всеки симулационен цикъл. Като резултат, налице е ситуация на взаимна блокировка в процеса на симулация.

ПРЕДЛАГАН ОПЕРАЦИОНЕН МОДЕЛ

В настоящата статия се предлага операционен модел, чрез който се цели намаляване на споменатите разходи. Моделът съдържа три компонента-комуникационен модул (КМ), синхронизационен модул (СМ) и симулационно ядро (СЯ), реализирани като отделно изпълняващи се нишки (фиг.2).



Фиг. 2. Предлаган операционен модел

Предназначението на комуникационния модул е аналогично на този от WARPED модела. Съществената разлика е отделянето на операциите по синхронизация в самостоятелен синхронизационен модул чрез което се цели действията по синхронизиране и обработка на събитията да се изпълняват паралелно. След завършване на синхронизацията, събитията ще бъдат депозирани от СМ в КСС за последваща обработка от СЯ.

Характерна особеност на модела е, че в случай на връщане назад СМ блокирва изпращането на всякакви съобщения до успешното възстановяване на предишното коректно състояние. Генерираните по време на фазата на обработка съобщения

няма бъдат изпратени и процесът на симулация ще продължи с възстановения КСС. Докато предходният модел позволява СО да се развиват прекалено далеч в моделното време между две връщания, предлаганият операционен модел ограничава тази възможност.

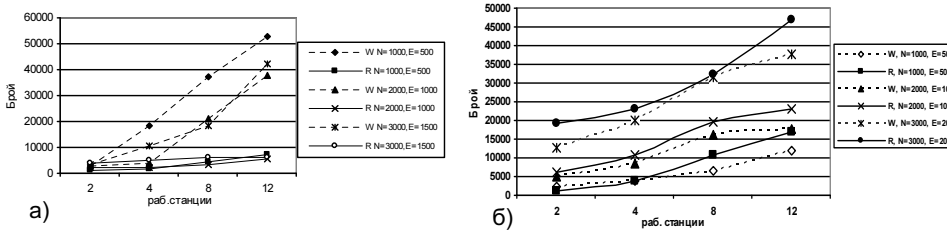
На базата на тези съображения могат да се направят изводи за положителните резултати от прилагането на този операционен модел:

- Моделът позволява отстраняването на коректността да бъде извършено в рамките само на един симулационен цикъл, независимо от броя на съобщенията-нарушители, получени през този цикъл;
- Намаляване на броя на изпращаните през мрежата анти-съобщения. Тъй като в случай на връщане назад се блокира възможността от изпращане на нормални съобщения в края на симулационния цикъл, отпада и необходимостта от анти-съобщения които да ги отменят;
- Намаляване на броя на повторно изпращаните нормални съобщения;
- Ограничаването на дълбочината на връщане назад ще рефлектира в намаляване на разходите за възстановяване на състоянието;
- Намаляване като цяло на времето за симулация.

ЕКСПЕРИМЕНТАЛНА ОЦЕНКА И АНАЛИЗ

Експерименталните изследвания са извършени на базата на тестовия модел PHOLD [1]. Това е един от най-често използваните тестови модели за оценка на производителността на разпределената симулация. PHOLD съдържа фиксиран брой симулационни обекти N, свързани в 2D тороидална мрежа и множество от задачи (съобщения) E, циркулиращи между тях. Като дестинация за всяко ново планирано събитие се избира с равна вероятност някой от неговите четири съседни в 2D топологията симулационни обекти.

Експериментите са проведени върху мрежа от Linux работни станции, свързани с 100Mb Ethernet. Като показатели за оценка са изследвани: брой изпратени анти-съобщения, брой връщания назад и общо време за симулация. Експериментите са извършени с нарастващи стойности на броя на СО и събития.

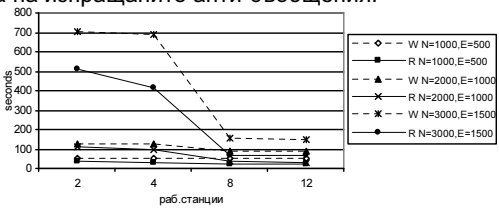


Фиг.3. Изпратени анти-съобщения (а) и брой връщания назад (б)

Представените на фиг.3-а резултати показват по-добрата ефективност на предлагания модел (R) по отношение на обема на генерираните анти-съобщения, изразяваща се в значителното му редуциране. Това е пряко следствие от паралелните действия по синхронизиране и обработка на събитията, при което благодарение на блокирането на изпращането на генерираните при обработката нормални съобщения в случай на връщане назад, се елиминира необходимостта от тяхното последващо отменяне чрез анти-съобщения. Същевременно следва да се отбележи, че нарастването на броя на анти-съобщенията (при увеличаване на броя на изчислителните възли) има по-плавен характер спрямо модела WARPED (W), независимо от нарастване размера на симулацията (броя на СО и събитията). Това се явява предпоставка за намаляване на комуникационните разходи, типични при наличието на голям брой работни станции.

От графиката на фиг.3-б се вижда обаче, че предлаганият модел допуска настъпването на по-голям брой връщания назад в сравнение с WARPED. Паралелното функциониране на синхронизационния модул и симулационното ядро на модела дава възможност действията по синхронизацията да се извършват едновременно с тези по обработката на събитията, което рефлектира в нарастване на броя на връщанията.

Въпреки това, като цяло при предлагания модел се наблюдава значително намаляване на общото време за симулация (фиг.4). Тези показатели се дължат основно на значителното редуциране на комуникационните разходи вследствие намаляването на обема на изпращаните анти-съобщения.



Фиг. 4. Време за изпълнение

ЗАКЛЮЧЕНИЕ

Използването на компютърните мрежи като изпълнителна платформа се явява алтернатива на специализираните сложни и скъпи платформи за разпределена симулация. В настоящата статия е представен операционен модел, функциониращ върху мрежа от работни станции. На базата на експериментална оценка е доказана неговата ефективност в сравнение с традиционния модел, използван в редица съвременни среди за РДСС. Като насоки за бъдеща работа се предвижда изследването и адаптирането на предложения модел за реализация върху клъстер от работни станции.

Благодарности

Тази разработка е подкрепена финансово от проект ”Подкрепа на творческото развитие на докторанти, постдокторанти и млади учени в областта на компютърните науки”, BG051PO001-3.3.04/13, финансиран от ЕВРОПЕЙСКИ СОЦИАЛЕН ФОНД 2007-2013, ОПЕРАТИВНА ПРОГРАМА „РАЗВИТИЕ НА ЧОВЕШКИТЕ РЕСУРСИ”

ЛИТЕРАТУРА

[1] Fujimoto R.M. Parallel and Distributed Simulation. Wiley, 2000.

[2] J. Banks, J. S. Carson, B. L. Nelson. Discrete-Event System Simulation (5th ed.), Upper Saddle River, Prentice Hall, 2009.

[3] K. Parumalla. Principles of Advanced and Distributed Simulation. Simulation, April 2009, 85, 227-228.

[4] MPI Forum, MPI2: A message-passing interface standard. The International Journal of High Performance Computing Applications 12 (1-2), 1998, 1-299.

[5] Nance R., Sargent R. Perspectives on the Evolution of Simulation. Operations Research, INFORMS, 2002, vol.50, no.1, 161-172.

[6] S., C. S. Yeo. A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools. Soft. Pract. Exper., 2004, 653-673.

За контакти:

Гл. ас. д-р инж. Христо Георгиев Вълчанов, Катедра “Компютърни науки и технологии”, Технически университет - Варна, тел.: 052/383424, e-mail: hristo@tu-varna.bg

Докладът е рецензиран.

РУСЕНСКИ УНИВЕРСИТЕТ „АНГЕЛ КЪНЧЕВ"
UNIVERSITY OF RUSE „ANGEL KANCHEV"

ДИПЛОМА

Програмният комитет на
Научната конференция РУ&СУ'11
награждава с КРИСТАЛЕН ПРИЗ
"THE BEST PAPER"
д-р ХРИСТО ГЕОРГИЕВ ВЪЛЧАНОВ
автор на доклада
" Операционна среда за
разпределена симулация "

DIPLOMA

The Programme Committee of
the Scientific Conference RU&SU'11
Awards the Crystal Prize
"THE BEST PAPER"
to HRISTO GEORGIEV VALCHANOV, PhD
author of the paper
"RunTime System for Distributed Simulation"

РЕКТОР
RECTOR

доц. дтн Христо Белоев
Prof. DSc Hristo Beloev

29.10.2011