

## СИМУЛЯЦИЯ СЕТИ С ПЕРЕМЕННОЙ ТОПОЛОГИЕЙ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Кашанов И.В., Шамин П.Ю.

*Государственное образовательное учреждение высшего профессионального образования "Владимирский государственный университет"*

*г. Владимир, Россия*

## THE SIMULATION OF DYNAMIC TOPOLOGY NETWORKS USING PARALLEL COMPUTING

Kashanov I.V., Shamin P.U.

*Vladimir State University*

*Vladimir, Russia*

Исследование параметров компьютерных сетей при различных характеристиках отдельных компонентов позволяет выбрать сетевое и вычислительное оборудование с учетом производительности, качества обслуживания, надежности и стоимости. Поскольку стоимость одного порта активного сетевого оборудования в зависимости от производителя оборудования, используемой технологии, надежности, управляемости может меняться от десятков рублей до десятков тысяч, моделирование позволяет минимизировать стоимость оборудования, предназначенного для использования в компьютерных сетях.

Существует большое количество сетевых симуляторов, в пределах от очень простого к очень сложному, способных моделировать глобальные сети, с большим числом узлов (персональные компьютеры, сервера, маршрутизаторы, переключатели, концентраторы и др.) и разнообразным набором параметров. Моделирование таких сетей требует больших вычислительных мощностей, поэтому происходит на высокопроизводительных кластерных системах с применением параллельных вычислений.

Одна из возможных архитектур сетевых симуляторов применяющих параллельные вычисления – моделирование целой сети в каждом процессе. Согласно этой архитектуре общий процесс распараллеливания можно представить в виде 3 шагов (рис. 1).

На первом шаге нулевой процесс генерирует сеть, определяет топологию, разнообразные параметры. Затем рассылает полученную сеть всем остальным процессам.

В зависимости от числа узлов, на каждом процессе определяется свой диапазон «активных» узлов. Это значит, что узлы только из этого диапазона могут выполнять все необходимые функции (зависят от целей моделирования; например, генерация сообщений), остальные могут лишь принимать и передавать сообщения.

На втором шаге моделируется работа сети, при этом каждый процесс выполняет поставленные задачи только над «активными» узлами.

По завершению всех операций, на третьем шаге, все полученные данные посылаются нулевому процессу, который их обрабатывает и выводит результаты.

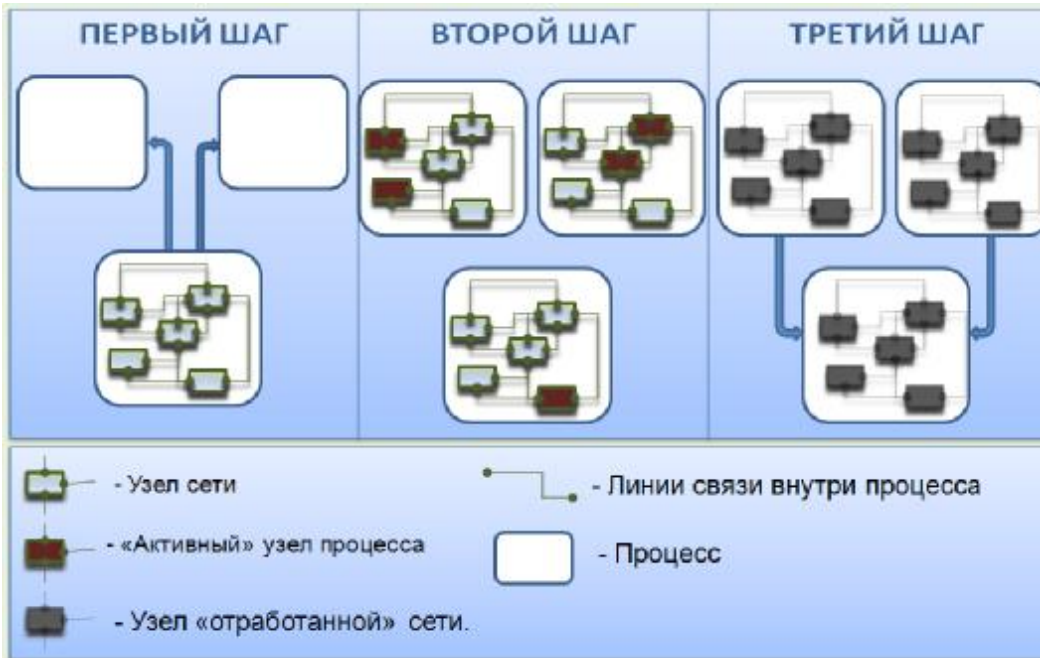


Рис. 1

Вторая архитектура сетевого симулятора с применением параллельных вычислений – моделирование фрагмента сети в каждом процессе. Данную архитектуру можно также представить в виде 3 шагов (рис. 2).

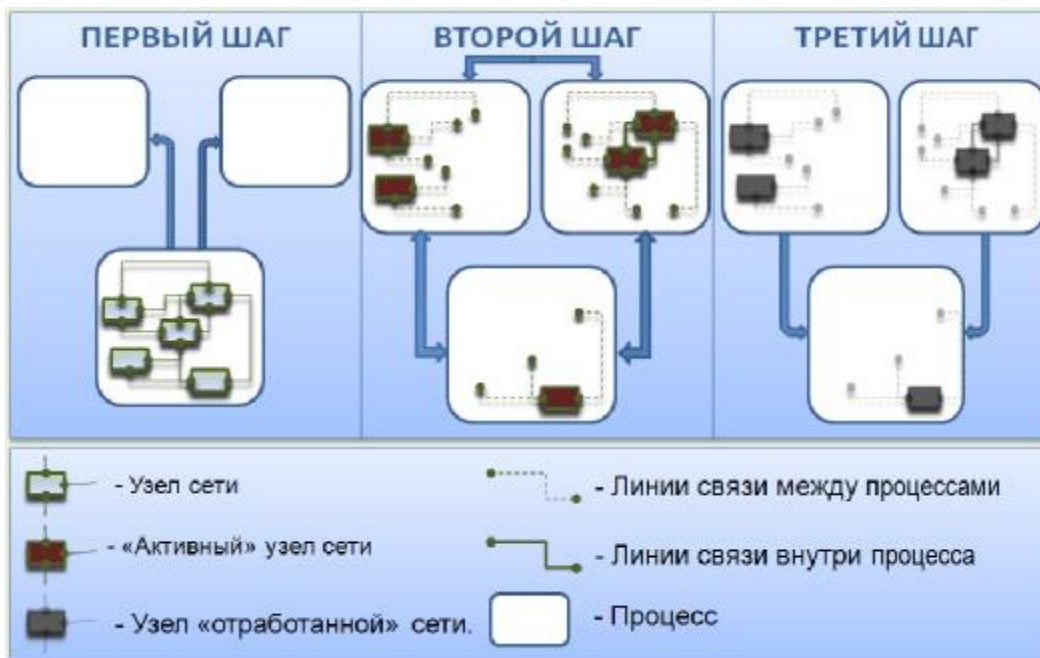


Рис. 2

На первом шаге нулевой процесс генерирует сеть, определяет топологию, разнообразные параметры. Затем каждому процессу посылает фрагмент сети – диапазон «активных» узлов.

На втором шаге моделируется работа сети, при этом каждый процесс оперирует со своим фрагментом сети и с фрагментами других процессов по каналам связи между процессами.

На третьем шаге происходит сбор полученной информации, ее анализ и вывод результатов.

Каждая из описанных архитектур сетевого симулятора, применяющего параллельные вычисления, имеет как преимущества, так и недостатки. Главным недостатком первой архитектуры является то, что требуются большие объемы памяти, так как каждый процесс хранит всю моделируемую сеть. К недостаткам второй архитектуры относится использование каналов связи между процессами, что замедляет работу симулятора.

Сравнительные испытания проводились на кластерах с использованием собственного, простого симулятора, способного генерировать сеть, определять топологию и имитировать передачу простого сообщения, характеризующегося временем жизни – TTL.

В ходе испытаний выяснилось, что при написании сетевого симулятора с использованием параллельных вычислений эффективней использовать первую архитектуру – модель целой сети в каждом процессе. Выбор этой архитектуры обоснован тем, что, хотя и расходуется значительный объем памяти, передача сообщений между узлами моделируемой сети внутри одного процесса проходит быстрее, чем между узлами разных процессов.