

# Архитектура приложения для исполнения на нескольких рабочих станциях, построенная с использованием модели GraphPlus

И.Д. Зекцер

Модель GraphPlus относится к инструментам для решения задач высокой вычислительной сложности [2]. Однако все вычисления, выполняемые с использованием данной модели в текущей реализации, производятся в рамках одной рабочей станции, а увеличение производительности достигается путем многопоточного исполнения на нескольких процессорах или в пакетном режиме [1].

Согласно [1], система, строящаяся по модели GraphPlus, состоит из трех структурных элементов: класс *Постояльцев (Resident)*, класс *Посетителей (Visitor)* и класс *Комнат (Room)*. Таким образом, имеется несколько объектов-комнат. В каждой из комнат может находиться один или несколько объектов-постояльцев. Объект-постоялец может быть посещен объектом-посетителем. При посещении происходит изменение внутреннего состояния пары объектов-постоялец-посетитель [1].

В результате анализа данной модели было выделено соответствие между объектом-комнатой и узлом вычислительной сети. Один или более объектов-комнат можно расположить на одной рабочей станции. Объекты-постояльцы, расположенные в комнатах, находящихся на разных узлах системы, будут взаимодействовать посредством принципа удаленного вызова процедур, а объекты-постояльцы, расположенные в одних и тех же комнатах, а также в комнатах, расположенных на одном узле сети, будут вызывать методы друг друга стандартным образом. Предлагается, чтобы каждый узел системы знал обо всех имеющихся в системе объектах-комнатах, объектах-постояльцах и объектах-посетителях, т.е. вводится избыточность, что сделано в целях обеспечения свойства отказоустойчивости. В определенный момент времени на том или ином узле системы будут активны только те объекты-постояльцы, которые принадлежат комнате, зарегистрированной на данном узле, и являются для данного узла активными комнатами. Таким образом, вводятся понятия *Активная Комната* и *Неактивная Комната*, *Активный Объект-Постоялец* и *Неактивный Объект-Постоялец*.

На каждом узле системы для каждого объекта-постояльца заводится объект класса *Заместитель (Proxy)*. Таким образом, при вызове в рамках данного узла метода неактивного объекта-постояльца происходит в действительности вызов метода его объекта-заместителя на данном узле системы. При попытке вызова метода объекта-заместителя он устанавливает соединение с удаленным объектом-постояльцем, перенаправляет ему данный вызов метода, а также передает соответствующие параметры. Поскольку дальнейший план разработок предполагает, что данная модель будет адаптироваться под состояние вычислительных узлов сети, то объекты-постояльцы могут мигрировать между комнатами, следовательно, их состояние может переходить из неактивного в активное и наоборот.

Кроме класса *Заместитель*, в усовершенствованной архитектуре модели GraphPlus вводится сервисный класс *Балансировщик (Balancer)*. С его помощью производится регистрация комнат на узлах, а также регистрация и распределение объектов-постояльцев по комнатам.

Дальнейшие планы по разработке данной архитектуры - реализация механизма обеспечения отказоустойчивости; тестирование увеличения производительности системы за счет предложенных усовершенствований; исследование различных алгоритмов балансировки.

## Литература

1. Востокин С.В. Применение метода парного взаимодействия объектов для построения сред разработки распределенных приложений. — Вестник СамГТУ, 2005, №38, с.26-28.
2. Востокин С.В. Применение интерпретатора сценария GraphPlus для управления распределенными вычислениями. — Известия СибИ РАН, том 7, №1(13) 2005, с.138-142.