тодологии Agile команда разработчиков должна состоять из высококвалифицированных специалистов, уход любого из которых наносит большой ущерб разработке проекта.

ВЫВОДЫ

Проведен качественный анализ рисков в зависимости от выбранной модели ПП и разработаны рекомендации по реагированию на выявленный риск.

Для RUP наиболее критичным оказался риск нарушения календарного планирования, для MSF – изменение требований, для Agile – текучесть кадров. Предложенные реакции позволят снизить негативное влияние рисков на конечные цели проектов и позволят выполнить проект в установленные сроки с заданным бюджетом.

СПИСОК ЛИТЕРАТУРЫ

1. Microsoft Solutions Framework. Дисциплина управления рисками MSF, вер. 1.1: [Электрон. ресурс]. – Режим доступа: http://www.microsoft.com/rus/msf.
2. PMBOK Руководство к Своду знаний по управлению проектами, 3-е изд. / PMBOK, Project Management Institute. – PMI, 2004. – 411 с.
3. *Груздо, И. В.* Повышение качества программного проекта за счет управления рисками / И. В. Груздо // Моделювання в економіці, організація виробництва та управління проектами. – 2009. – 1. – С. 141–146.
4. *ДеМарко, Т.* Вальсируя с медведями Управление рисками в проектах по разработке программного обеспечения / Том ДеМарко, Тимоти Листер, p.m. Office, 2005. – 190 с.
5. *Брагина, Т. И.* Сравнительный анализ итеративных моделей разработки программного обеспечения / Т. И. Брагина, Г. В. Табунщик // Радиоелектроніка, інформатика, управління. – 2010. – № 2. – С. 130–139.
6. *Bragina, T.* Comparative Analysis of Software Development Models for Electro-technical Systems / T. Bragina, G. Tabunshchik // Proc. of Int. Conf. on Modern Problem of Radio Engineering, Telecommunications and Computer Science TCSET'2010, February 19–23, 2010, Lviv–Slavsko, Ukraine. – P. 347.
7. *Брагина, Т. И.* Классификация моделей итеративной разработки программного обеспечения / Т. И. Брагина, Г. В. Табунщик // Системный анализ. Информатика. Управление : материалы Всеукраинской научно-практической конференции, САИУ–2010, Март 04–05, 2010, Запорожье, Украина. – Запорожье : КПУ, 2010. – С. 23–25.
8. *Галатенко, В. А.* Управление рисками: обзор употребительных подходов / Галатенко В.А. // Информационный бюллетень. – №11(162). – 2006. – С. 1–15
9. *Липаев, В. В.* Анализ и сокращение рисков проектов программных средств / В.В. Липаев // Jet Info. – 2005. – №1. – С. 1–36.
10. *Астахов, А.* Как управлять рисками информационной безопасности? / А. Астахов // CISA. – ноябрь 2006. – С.121 – 124.
11. *Петренко, С.* Методики и технологии управления информационными рисками / С. Петренко, С. Симонов // IT Manager. – 2003. – №3. – С. 57–61.

Стаття надійшла до редакції 09.11.2010.
Після доробки 11.03.2011.

Брагіна Т. І., Табунщик Г. В.

АНАЛІЗ ПІДХОДІВ ДО КЕРУВАННЯ РИЗИКАМИ В ПРОГРАМНИХ ПРОЕКТАХ З ІТЕРАЦІЙНИМ ЖИТТЄВИМ ЦИКЛОМ

Виконано огляд головних ризиків програмних проектів з ітераційним життєвим циклом. Запропоновані рекомендації з управління виявленими ризиками в залежності від обраної моделі розробки програмних проектів. Виділено найбільш критичні ризики та проведено їх якісний аналіз.

Ключові слова: керування ризиками, програмні проекти, реакція на ризик.

Bragina T. I., Tabunshchyk G.V.

ANALYSIS RISK MANAGEMENT APPROACHES IN SOFTWARE PROJECTS WITH AN ITERATIVE LIFECYCLE

In the article there is performed a review of the main risks in the software projects with an iterative lifecycle. The authors developed guidelines for the risks identified management, depended on the software development projects model. The most critical risks are identified and the qualitative analysis was made by the authors.

Key words: the risk management, the software projects, reaction on the risk.

УДК 004.75

Дьячук Т. С.

Асистент Запорожского национального технического университета

ОЦЕНКА ХАРАКТЕРИСТИК РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

Выполнен обобщенный анализ распределенной системы с точки зрения систем массового обслуживания. Определены характеристики, которые могут быть исследованы с помощью систем массового обслуживания.

Ключевые слова: система массового обслуживания, Grid, кластер, ресурс, задача.

ВВЕДЕНИЕ

Оптимизация ресурсов является одной из важных задач в современном мире. Существует множество видов ресурсов: материальные, информационные, финансовые, интеллектуальные, трудовые, энергетические, времени и другие. Эффективное использование ресурсов

[1] в сфере высокопроизводительных вычислений не является исключением, так как вычислительная нагрузка на информационные системы постоянно растет. Получить максимальный результат при минимальных затратах стремятся как владельцы ресурсов, так и пользователи этих ресурсов. Для облегчения проектирования, контроля, планирования использования и анализа эффективности распределенной системы стремятся свести все измерители ресурсов к одному – например, денежному. При этом возникают трудности из-за отсутствия методов оценки некоторых видов ресурсов в денежном выражении. Поэтому при оценке эффективности использования ресурсов применяются несколько показателей, в совокупности отражающих уровень потребления ресурсов. В данной работе рассматривается распределенная система с помощью математического аппарата систем массового обслуживания. Данная модель является упрощенной в связи со сложностью реальных систем, но в целом позволяет получить характеристики, которые можно использовать для анализа свойств реальных распределенных систем.

ПОСТАНОВКА ЗАДАЧИ И ЦЕЛЬ РАБОТЫ

В настоящее время широко используется идея объединения ресурсов различных организаций в одну распределенную вычислительную систему в рамках Grid -проектов. В связи с чем возникает актуальная задача оптимального развития Grid, которая заключается в повышении производительности системы с одной стороны и минимизации затрат на сопровождение инфраструктуры с другой. В Grid -системе существуют два уровня планирования [2]: глобальный – распределение задач между кластерами (объединение машин в рамках одной организации) и локальный – распределение внутри кластера.

Пользовательские задачи распределяются между кластерами, учитывая информацию о задачах и текущем состоянии кластеров. Считаем, что одна задача выполняется на одном кластере, то есть не разбивается по нескольким. Подбирается кластер, с подходящими по характеристикам машинами и позволяющий решить задачу за минимальное время.

Анализируется запрос на выполнение задачи и рассылается всем кластерам, которые могут его выполнить. Каждый кластер на основании информации, содержащейся в запросе, своей текущей загрузки и уже составленного расписания возвращает время, до которого задача может быть выполнена. Принимается решение на основе этой информации и выбирается кластер, который удовлетворяет предельному времени выполнения задачи и имеет наименьшую стоимость (или удовлетворяет предельной стоимости и имеет минимальное время выполнения). Затем с выбранным кластером заключается соглашение и посылается задача на выполнение. Иерархическая система распределения задач имеет два уровня:

– распределение задач по кластерам (обслуживается брокерами);

– распределение многопроцессорных задач на кластере (обслуживается локальными планировщиками кластеров).

Миграция задач с одного кластера на другой также влияет на эффективность распределения. Задача может быть возвращена брокеру (с некоторым штрафом), если кластер получит больше выгодных задач от других брокеров или в случае сбоя. Брокер перераспределяет возвращенную задачу на другой кластер.

Выбор ресурсов и распределение вычислительной нагрузки осуществляется таким образом, чтобы удовлетворить все требования пользователей и оптимизировать общее время выполнения и стоимость используемых ресурсов.

В данной работе предложена упрощенная модель оценки характеристик кластера рабочих станций.

МОДЕЛЬ ЗАГРУЗКИ КЛАСТЕРА

Рассмотрим процесс решения задач на кластере с точки зрения систем массового обслуживания [3]. Загрузка и получение результатов на каждой рабочей станции представляют собой случайные события.

Будем считать, что вероятность получения результата на рабочей станции в малом промежутке времени Δt пропорциональна этому промежутку, то есть

$$\Delta p(t) = p(t+\Delta t) - p(t) = [1 - p(t)]\lambda\Delta t ,$$

где $p(t)$ – вероятность того, что за время t получено решение задачи; λ – коэффициент пропорциональности.

Выражение в квадратных скобках соответствует вероятности отсутствия решения в момент времени t.

При $\Delta t \to 0$ получим дифференциальное уравнение

$$dp(t) = [1 - p(t)]\lambda dt ,$$

откуда

$$\frac{dp(t)}{1-p(t)} = \lambda dt .$$

После интегрирования

$$\int_0^t \frac{dp(t)}{1-p(t)} = \int_0^t \lambda dt ,$$

получим

$$\ln[1-p(t)] = -\lambda t \ \text{ или } \ p(t) = 1 - e^{-\lambda t} .$$

Плотность распределения времени решения задачи соответственно равна

$$r(t) = \frac{dp(t)}{dt} = \lambda e^{-\lambda t} ,$$

то есть имеет экспоненциальный закон распределения.

В простейшем случае все рабочие станции имеют одинаковые характеристики, и плотности распределения решения будут соответственно одинаковы. Величина λ отражает обобщенные характеристики рабочей станции и представляет собой интенсивность решения задач на станции.

Рассмотрим поток задач, поступающих на кластер. Будем считать вероятность поступления задачи в бесконечно малом промежутке времени Δt также пропорциональна величине этого промежутка, то есть

$$\Delta s(t) = s(t+\Delta t) - s(t) = [1 - s(t)]\mu\Delta t\,,$$

где $s(t)$ – вероятность того, что за время t получено решение задачи; μ – коэффициент пропорциональности.

Выражение в квадратных скобках соответствует вероятности отсутствия задач в момент времени t.

При $\Delta t \to 0$ получим дифференциальное уравнение

$$ds(t) = [1 - s(t)]\mu dt\,,$$

откуда

$$\frac{ds(t)}{1-s(t)} = \mu dt\,.$$

После интегрирования

$$\int_0^t \frac{ds(t)}{1-s(t)} = \int_0^t \mu dt\,,$$

получим

$$\ln[1-s(t)] = -\mu t \text{ или } s(t) = 1 - e^{-\mu t}\,.$$

Плотность распределения времени поступления задачи соответственно равна

$$v(t) = \frac{ds(t)}{dt} = \mu e^{-\mu t}\,,$$

то есть также имеет экспоненциальный закон распределения.

На рис. 1 приведен граф состояний системы.

Рассмотрим состояния рабочих станций кластера:

S_0 – состояние, когда все рабочие станции загружены решением задач;

S_1 – состояние, при котором на одной из рабочих станций завершено решение;

……………………………………

S_i – состояние, когда решение завершено на i рабочих станциях;

……………………………………

S_n – состояние завершения решений на всех рабочих станциях.

Обозначим через $p_0(t), p_1(t), \ldots, p_i(t), \ldots, p_n(t)$ вероятности нахождения системы в соответствующих состояниях. Тогда можно записать следующие дифференциальные уравнения

$$\frac{dp_0(t)}{dt} = -\lambda_{0.1}p_0(t) + \mu_{1,0}p_1(t);$$
$$\frac{dp_1(t)}{dt} = \lambda_{0,1}p_0(t) - (\mu_{1,0} + \lambda_{1,2})p_1(t) + \mu_{2,1}p_2(t);$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\frac{dp_i(t)}{dt} = \lambda_{i-1,i}p_{i-1}(t) - (\mu_{i,i-1} + \lambda_{i,i+1})p_i(t) + \mu_{i+1,i}p_{i+1}(t);$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\frac{dp_n(t)}{dt} = \lambda_{n-1.n}p_{n-1}(t) - \mu_{n,n-1}p_n(t);$$
$$\sum_{i=0}^{n} p_i = 1.$$

В граничном стационарном режиме при $t \to \infty$ будем иметь систему алгебраических уравнений

$$0 = -\lambda_{0.1}p_0(t) + \mu_{1,0}p_1(t);$$
$$0 = \lambda_{0,1}p_0(t) - (\mu_{1,0} + \lambda_{1,2})p_1(t) + \mu_{2,1}p_2(t);$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$0 = \lambda_{i-1,i}p_{i-1}(t) - (\mu_{i,i-1} + \lambda_{i,i+1})p_i(t) + \mu_{i+1,i}p_{i+1}(t);$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$0 = \lambda_{n-1.n}p_{n-1}(t) - \mu_{n,n-1}p_n(t);$$
$$\sum_{i=0}^{n} p_i = 1.$$

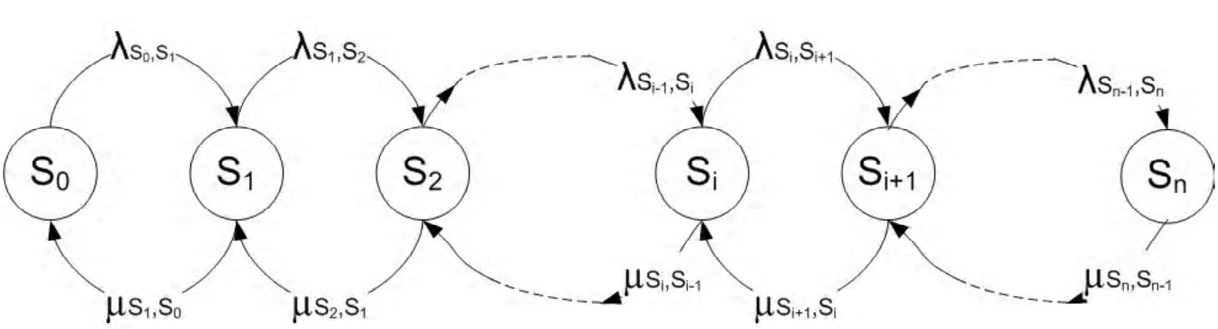

Рис. 1. Граф состояний системы

Для кластера, состоящего из n рабочих станций значения $\lambda_{i-1,i}$ и $\mu_{i,i-1}$ соответственно равны

$$\lambda_{i,i+1}=(n-i)\lambda;\quad \mu_{i+1,i}=(i+1)\mu,\quad i=0,\ldots,n-1.$$

Ожидаемое количество свободных рабочих станций кластера $S_{\text{св}}$ и средняя загрузка рабочих станций $S_{\text{заг}}$ равны соответственно

$$S_{\text{св}}=1p_1+2p_2+\ldots+np_n;$$
$$S_{\text{заг}}=np_0+(n-1)p_1+\ldots+p_{n-1}.$$

Справедливо также следующее утверждение $S_{\text{заг}}=n-S_{\text{св}}$, так как

$$S_{\text{св}}+S_{\text{заг}}=1p_1+2p_2+\ldots+(n-1)p_{n-1}+np_n+$$
$$+np_0+(n-1)p_1+\ldots+p_{n-1}=n\cdot\sum_{i=0}^{n}p_i=n\cdot 1=n.$$

Готовность системы к принятию задач для решения равна

$$G=p_1+p_2+\ldots+p_n=1-p_0.$$

Вероятность отказа в обслуживании:

$$P_{\text{отк}}=p_0.$$

Оценим значения вероятностей нахождения системы в различных состояниях. Для этого представим систему уравнений в матричном виде, убрав одно из равенств, в котором нет необходимости:

$$M\cdot P=I,$$

где

$$M=\begin{bmatrix}
-n\lambda & \mu & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0\\
n\lambda & -(n-1)\lambda-\mu & 2\mu & \cdots & 0 & 0 & \cdots & 0 & 0 & 0\\
0 & (n-1)\lambda & -(n-2)\lambda-2\mu & \cdots & 0 & 0 & \cdots & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & \cdots & -(n-i+1)\lambda-i\mu & (i+1)\mu & \cdots & 0 & 0 & 0\\
0 & 0 & 0 & \cdots & (n-i+1)\lambda & -(n-i)\lambda-(i+1)\mu & \cdots & 0 & 0 & 0\\
0 & 0 & 0 & \cdots & 0 & (n-i)\lambda & \cdots & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2\lambda & -\lambda-(n-1)\mu & n\mu\\
1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1
\end{bmatrix};\ P=\begin{bmatrix}p_0\\ p_1\\ p_2\\ \vdots\\ p_{i-1}\\ p_i\\ p_{i+1}\\ \vdots\\ p_{n-1}\\ p_n\end{bmatrix};\ I=\begin{bmatrix}0\\0\\0\\ \vdots\\0\\0\\0\\ \vdots\\0\\1\end{bmatrix}. \quad (1)$$

Определим значение вероятности p_i с помощью определителей. Учитывая особенности матрицы, отметим следующее свойство определителей: определитель треугольной матрицы равен произведению диагональных элементов. Кроме того, покажем, что произведению диагональных элементов матрицы будет иметь определитель вида:

$$\Delta=\begin{vmatrix}
a_{11} & b_{12} & 0 & \vdots & 0 & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0\\
0 & a_{22} & b_{23} & \vdots & 0 & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots\\
0 & 0 & 0 & \vdots & a_{i-2,i-2} & b_{i-2,i-1} & 0 & 0 & 0 & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & 0 & a_{i-1,i-1} & b_{i-1,i} & 0 & 0 & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & 0 & 0 & a_{ii} & 0 & 0 & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & a_{i+1,i+1} & 0 & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & b_{i+2,i+1} & a_{i+2,i} & \vdots & 0 & 0 & 0\\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & b_{i+3,i} & \vdots & 0 & 0 & 0\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & 0 & \vdots & a_{n-2,n-2} & 0 & 0\\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & 0 & \vdots & b_{n-1,n-2} & a_{n-1,n-1} & 0\\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & 0 & \vdots & 0 & b_{n,n-1} & a_{nn}
\end{vmatrix}. \quad (2)$$

Нетрудно видеть, что последовательное разложение определителей по элементам столбцов, начиная с последнего столбца до столбца ($i+1$), которые представляют собой диагональные элементы, приведет к треугольной матрице, значение которой будет также произведение диагональных элементов. В результате получим:

$$\Delta = \prod_{i=1}^{n} a_{ii} .$$

Найдем определитель Δ матрицы М системы. Для этого преобразуем матрицу складыванием предыдущей строки с каждой последующей за исключением последней. В результате получим

$$\det M = \Delta = \begin{vmatrix} -n\lambda & \mu & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -(n-1)\lambda & 2\mu & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -(n-2)\lambda & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -(n-i+1)\lambda & (i+1)\mu & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -(n-i)\lambda & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -\lambda & n\mu \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \end{vmatrix}.$$

Выполним разложение определителя по элементам нижней строки. Алгебраические дополнения первого и последнего элементов этой строки представляют собой треугольные матрицы, алгебраические дополнения остальных элементов представляют определители вида (2).

Алгебраическое дополнение элемента ($n+1$) строки и ($n+1$) столбца представляет собой треугольную матрицу $n \times n$, диагональ которой содержит все отрицательные элементы. Значение определителя будет иметь знак положительный, если n имеет четное значение, и отрицательный, если n – нечетное. С учетом значений элементов диагонали будем иметь $(-1)^n n!\lambda^n$.

Рассмотрим алгебраические дополнения элементов ($n+1$) строки и i-го столбца. Знак алгебраического дополнения для элементов ($n+1$) строки и i-го столбца будет определяться степенью $(-1)^{n+1+i}$. С другой стороны, определитель будет иметь вид матрицы (2), причем значения элементов диагонали столбцов от 1 до (i–1) будет отрицательным, а значения элементов диагонали столбцов от i до n – положительным. Следовательно, знак определителя минора определится степенью $(-1)^{i-1}$. Отсюда знак алгебраического дополнения будет определяться произведением $(-1)^{n+1+i} \cdot (-1)^{i-1} = (-1)^{n+2i}$, то есть если n имеет четное значение, то положительный и отрицательный, если n – нечетное.

С учетом значений элементов диагонали будем иметь $(-1)^n \frac{n!}{i!}\lambda^{n-i} \frac{n!}{(n-i)!}\mu^i = (-1)^n n! C_n^i \lambda^{n-i}\mu^i$, где C_n^i – число сочетаний из n по i. Отсюда имеем

$$\Delta = (-1)^n n! \sum_{i=0}^{n} C_n^i \lambda^{n-i}\mu^i = (-1)^n n!(\lambda+\mu)^n,$$

где $\sum_{i=0}^{n} C_n^i \lambda^{n-i}\mu^i = (\lambda+\mu)^n$ согласно биному Ньютона.

При нахождении определителей Δ_i, у которых столбец i заменен на вектор свободных членов, который представлен 1 в ($n+1$) строке и нулями для остальных строк, его значение и знак будет таким же, как и для алгебраического дополнения элементов ($n+1$) строки и i столбца.

$$\Delta_i = \begin{vmatrix} -n\lambda & \mu & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ n\lambda & -(n-1)\lambda-\mu & 2\mu & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & (n-1)\lambda & -(n-2)\lambda-2\mu & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -(n-i+1)\lambda-i\mu & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & (n-i+1)\lambda & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2\lambda & -\lambda-(n-1)\mu & n\mu \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \end{vmatrix}.$$

Складывая предыдущую строку с каждой последующей, за исключением последней, получим

$$\Delta_i = \begin{vmatrix} -n\lambda & \mu & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -(n-1)\lambda & 2\mu & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -(n-2)\lambda & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -(n-i+1)\lambda & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -\lambda & n\mu \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \end{vmatrix}.$$

После разложения по элементам столбца i получим

$$\Delta_i = (-1)^n n! C_n^i \lambda^{n-i} \mu^i .$$

Отсюда

$$p_i = \frac{\Delta_i}{\Delta} = C_n^i \frac{\lambda^{n-i}\mu^i}{(\lambda+\mu)^n} .$$

Оценим ожидаемое количество свободных станций $S_{\text{св}}$ для загрузки новых заданий:

$$S_{\text{св}} = \sum_{i=1}^{n} i C_n^i \frac{\lambda^{n-i}\mu^i}{(\lambda+\mu)^n} .$$

Так как $C_n^i = \frac{n}{i} C_{n-1}^{i-1}$, то можно записать

$$S_{\text{св}} = \sum_{i=1}^{n} i \frac{n}{i} C_{n-1}^{i-1} \frac{\lambda^{n-i}\mu^i}{(\lambda+\mu)^n} = \frac{n\mu}{(\lambda+\mu)} \sum_{i=1}^{n} C_{n-1}^{i-1} \frac{\lambda^{n-i}\mu^{i-1}}{(\lambda+\mu)^{n-1}}.$$

Выражение под знаком суммы представляет собой сумму вероятностей всех состояний системы для количества станций, равном (n–1), то есть равно единице. В результате получим

$$S_{\text{св}} = \frac{n\mu}{(\lambda+\mu)}.$$

Количество загруженных станций $S_{\text{заг}}$ равно

$$S_{\text{заг}} = n - S_{\text{св}} = n - \frac{n\mu}{\lambda+\mu} = \frac{n\lambda}{\lambda+\mu}.$$

Готовность системы для принятия новых задач определяется вероятностью

$$G = 1 - \frac{\lambda^n}{(\lambda+\mu)^n}.$$

Вероятность отказа в обслуживании:

$$P_{\text{отк}} = \frac{\lambda^n}{(\lambda+\mu)^n}.$$

Для любой системы массового обслуживания, в которой заявка может обслуживаться одним каналом, среднее число заявок, обслуживаемых в единицу времени, определяется как произведение числа занятых каналов на плотность потока обслуживания. В нашем случае, количество задач K, решаемых в единицу времени, определится произведением $S_{\text{заг}}$ на величину λ. Таким образом, получим

$$K = S_{\text{заг}}\lambda = \frac{n\lambda^2}{\lambda+\mu} .$$

Рассмотрим случай разбиения каждой задачи на подзадачи, которые могут выполняться параллельно на различных рабочих станциях. Пусть каждая задача представлена в виде m подзадач, имеющих примерно одинаковые характеристики. Среднее время решения подзадачи будет в m раз меньше и, следовательно, интенсивность ее решения $\lambda_{n/з}$ будет в m раз больше, то есть

$$\lambda_{n/з} = m\lambda .$$

Поток $\mu_{n/з}$ поступающих подзадач также увеличится в m:

$$\mu_{n/з} = m\mu .$$

Количество подзадач, решаемых в единицу времени определится выражением

$$K_{n/з} = \frac{n\lambda_{n/з}^2}{\lambda_{n/з} + \mu_{n/з}} = \frac{n(m\lambda)^2}{m\lambda + m\mu} = \frac{nm\lambda^2}{\lambda+\mu} .$$

Если в кластере одновременно решается M задач, имеющих одинаковые приоритеты, то среднее количество подзадач, относящихся к одной задаче, решаемых в единицу времени, равно

$$k = \frac{K_{n/з}}{M} = \frac{nm\lambda^2}{(\lambda+\mu)M}.$$

Среднее время решения задачи равно

$$\bar{t} = \frac{m}{k} = \frac{m(\lambda+\mu)M}{nm\lambda^2} = \frac{(\lambda+\mu)M}{n\lambda^2}.$$

Если поступление на решение новых подзадач рассматривать как следствие получения очередного решения, то $\lambda = \mu$. Таким образом, поток запросов на кластер меняется во времени в зависимости от интенсивности решения задач. Чем быстрее решаются задачи, тем сильнее увеличивается поток задач на глобальном уровне. Если есть простаивающие станции, то поток задач увеличивается. Среднее время на решение задачи определится выражением

$$\bar{t} = \frac{2M\lambda}{n\lambda^2} = \frac{2M}{n\lambda}.$$

Отсюда вытекает очевидный факт, что ожидаемое время на решение задачи прямо пропорционально количеству задач в системе и обратно пропорционально количеству рабочих станций в кластере.

Пусть каждая задача представлена различным количеством подзадач m_i, тогда общее количество подзадач равно

$$m_\Sigma = \sum_{i=1}^{M} m_i .$$

Будем считать, что подзадачи имеют одинаковые характеристики с точки зрения времени их решения, то есть интенсивность решения одинакова:

$$\forall(i=\overline{1,M})(\frac{\lambda_i}{m_i} = \lambda_{n/з} = \mathrm{const}),$$

где λ_i – интенсивность решения задачи i, величина обратная ожидаемому времени решения.

В этом случае среднее количество подзадач, решаемых в единицу времени для каждой задачи и определится стратегией планирования. Для случая, когда планирование осуществляется между подзадачами независимо от того, к каким задачам они относятся, среднее количество подзадач, решаемых в единицу времени, распределится пропорционально количеству подзадач в каждой задаче:

$$k_i = \frac{K_{n/з} m_i}{m_\Sigma},$$

а ожидаемое время решения задачи равно

$$\bar{t_i} = \frac{m_i}{k_i} = \frac{m_i m_\Sigma}{K_{n/з} m_i} = \frac{m_\Sigma}{K_{n/з}}.$$

Для данного случая время решения каждой задачи одинаковое.

Для кластера ожидаемое количество загруженных рабочих станций в этом случае будет иметь вид

$$S_{заг} = \frac{n\lambda_{n/з}}{\lambda_{n/з} + \mu_{n/з}},$$

а среднее количество подзадач, решаемых в единицу времени равно

$$K_{n/з} = S_{заг}\lambda_{n/з} = \frac{n\lambda_{n/з}^2}{\lambda_{n/з} + \mu_{n/з}}.$$

Если планирование осуществляется между задачами независимо от их объема (одни и те же ресурсы выделяются каждой задаче), то среднее время решения i-той задачи равно

$$\bar{t_i} = \frac{m_i}{K_{n/з}} = \frac{m_i(\lambda_{n/з} + \mu_{n/з})}{n\lambda_{n/з}^2}.$$

Из полученного выражения следует, что более короткие задачи будут решаться быстрее.

Оценим стоимость выполнения i-ой задачи

$$\mathrm{Cost}_i = \bar{t_i} \cdot g,$$

где g – стоимость использования ресурсов кластера в единицу времени.

ВЫВОДЫ

В работе предложена упрощенная модель оценки характеристик кластера рабочих станций. Данная модель отражает основные закономерности работы многопроцессорной системы. Результаты исследований могут быть применены администраторами при настройке оборудования кластерных систем для повышения эффективности их использования. Система массового обслуживания аналогичного типа приведена в [3], только в нашем случае применена другая трактовка и расширены исследуемые характеристики системы.

Часто кластеры строятся из однородных элементов, поэтому принято допущение, что все рабочие станции кластера имеют приблизительно одинаковые характеристики. Если рассматривать случай, когда кластер строится на основе компьютеров с разными характеристиками, то количество состояний резко возрастает, и системы уравнений становятся очень громоздкими, что не позволяет получить закономерности в общем случае.

В будущей работе планируется исследовать поступление задач с разными характеристиками, а также рассмотреть возможность назначения приоритетов. Дальнейшее развитие связано также с объединением кластеров на глобальном уровне для изучения их взаимодействия.

СПИСОК ЛІТЕРАТУРИ

1. *Дьячук, Т. С.* Дослідження ефективності роботи алгоритму планування ресурсів для розподіленої системи / Т. С. Дьячук, Р. К. Кудерметов, А. М. Літава, Р. І. Сметанін // Вісник КДПУ імені Михайла Остроградського. – Кременчук : КДУ, 2010. – Вип. 1/2010 (60) частина 1. – С. 47–50.
2. Эвристики распределения задач для брокера ресурсов Grid [Электронный ресурс] / А. И. Аветисян, С. С. Гайсарян, Д. А. Грушин, Н. Н. Кузюрин, А. В. Шокуров // Труды Института системного программирования РАН – *2004.* – Режим доступа: http://citforum.univ.kiev.ua/nets/digest/grid/index.shtml – Загл. с экрана.
3. *Клейнрок, Л.* Теория массового обслуживания. Пер. с англ. / Клейнрок, Л. ; пер. И. И. Грушко; ред. В.И. Неймана. – М. : Машиностроение, 1979. – 432 с.

Стаття надійшла до редакції 17.03.2011.

Дьячук Т. С.

ОЦІНКА ХАРАКТЕРИСТИК РОЗПОДІЛЕНОЇ СИСТЕМИ

Виконано узагальнений аналіз розподіленої системи з точки зору систем масового обслуговування. Визначено характеристики, які можуть бути досліджені за допомогою систем масового обслуговування.

Ключові слова: система масового обслуговування, Grid, кластер, ресурс, завдання.

Diachuk T. S.

EVALUATION OF DISTRIBUTED SYSTEM’S CHARACTERISTICS

The distributed system in terms of queueing systems is analyzed. The characteristics that can be studied using queuing systems are presented.

Key words: queueing system, Grid, cluster, resource, task.

УДК 004.9:004.82

Гонтарь Н. А.[1], Кудерметов Р. К.[2]

[1]*Ассистент Запорожского национального технического университета*
[2]*Канд. техн. наук, профессор Запорожского национального технического университета*

РАЗРАБОТКА ОНТОЛОГИИ СИСТЕМНОГО ИНЖИНИРИНГА КОСМИЧЕСКИХ СИСТЕМ

В статье обоснована необходимость создания онтологии системного инжиниринга (СИ). Онтология СИ вместе с переходом к моделе-ориентированному СИ повысит эффективность разработки сложных систем. Предложен прототип онтологической модели, основанный на дескриптивной логике и формализованный с помощью языка OWL DL.

Ключевые слова: системный инжиниринг, онтология, MBSE, дескриптивная логика, OWL.

ВВЕДЕНИЕ

Согласно определению Международного Совета по Системной Инженерии (International Council on Systems Engineering, INCOSE) системный инжиниринг (СИ) – это междисциплинарный подход и средства для создания успешных систем [1, 2]. На протяжении всего жизненного цикла (ЖЦ) систем (концептуальное проектирование, разработка, изготовление, испытания, эксплуатация и утилизация) СИ фокусирует внимание разработчиков на глубоком анализе и отслеживании потребностей пользователей создаваемой системы и функциональных требованиях к ней. СИ интегрирует дисциплины и коллективы разработчиков в единое распределенное виртуальное пространство проекта, в котором согласованы процессы разработки и изготовления системы, взаимодействия разработчиков, применяемые ими понятия, правила, методы, инструменты и т. д.

СИ – итерационный процесс и каждое новое, принятое в ходе ЖЦ системы, решение отражается на конечном результате разработки. Поэтому важным является создание на основе современных информационных технологий благоприятной среды, отражающей текущие процессы СИ, множество моделей системы, возможные варианты решений и ошибки.

В одной из последних инициатив INCOSE – «Systems Engineering Vision 2020» декларируются принципы моделе-ориентированного подхода к СИ, который получил название MBSE (Model-based Systems Engineering) [3, 4]. MBSE – это формализованное применение моделирования для поддержки системных требований, разработки, анализа, верификации и валидации систем на фазе концептуального проектирования и далее при разработке и последующих фазах ЖЦ системы. Основными задачами MBSE являются:

– обеспечить согласованное взаимодействие между заказчиками и разработчиками систем, а также между коллективами разработчиков;

– уменьшить риски и неопределенности и контролировать их в течение всего ЖЦ системы;

– поддерживать процессы контроля границ системы, ее агрегации, декомпозиции, интеграции, сертификации и прогнозирования функциональных возможностей;

– обеспечить согласованность множества системных аспектов, видений и моделей системы;