

АНАЛИЗ ПРОБЛЕМЫ СИНТЕЗА МУЛЬТИВЕРСИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Васильев К.О.,

научный руководитель канд. тех. наук Царев Р. Ю.

Сибирский Федеральный Университет

Введение

При разработке программных систем, отказ которых может привести к существенному материальному ущербу или ущербу жизни и здоровью людей (области ракетно-космической техники, ядерного, химического производства, медицины и пр.), одной из самых важных характеристик следует считать надёжность программной системы.

Существует множество способов повысить надёжность программной системы. Одним из самых успешных является разработка с использованием мультиверсионного подхода, при котором для каждого модуля программной системы разрабатывается несколько версий его реализации, которые различаются степенью надёжности, затратами на выполнение (процессорным временем, необходимым объёмом памяти для выполнения) и т.д.

1. Цель создания мультиверсионного программного обеспечения

Мультиверсионное программное обеспечение создаётся с целью максимизации надёжности разрабатываемой системы, что математически может быть выражено, как:

$$\max R = \prod_{i=1}^n R_i$$

Где:

R – надёжность программной системы;

R_i – надёжность i -го модуля;

n – число модулей в системе.

При этом надёжность отдельного модуля можно вычислить, как:

$$R_i = 1 - \prod_{j=1}^{m_i} (1 - R_{ij})^{X_{ij}} \quad (1)$$

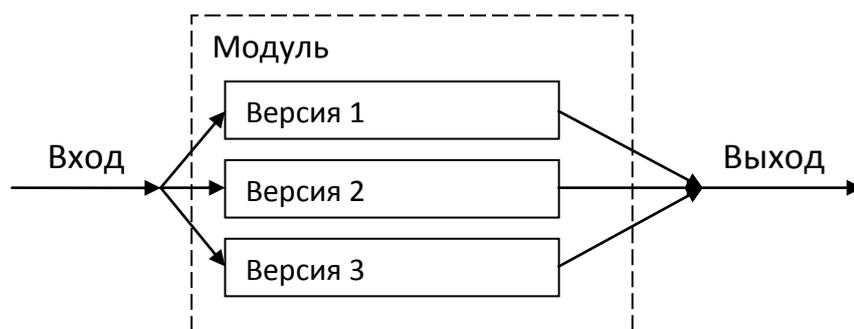
Где:

m_i – количество версий i -го модуля;

R_{ij} – надёжность j -й версии i -го модуля;

X_{ij} – дискретное число, равное 1, если j -я версия i -го модуля будет принята, и 0 в противном случае.

Покажем, что использование мультиверсионного программного обеспечения увеличивает надёжность программной системы на примере программного обеспечения, состоящего из одного модуля и трёх его версий.



Пусть надёжность будет определяться вероятностью безотказной работы и для каждой версии она будет следующей:

$$R_1 = 0,97$$

$$R_2 = 0,975$$

$$R_3 = 0,95$$

Данные показатели надёжности – хорошие для систем, к которым не предъявляется серьёзных требований, но достаточно плохие для высоконагруженных и дорогостоящих систем. Используя формулу (1), видно, что при использовании мультиверсионного программного обеспечения надёжность системы будет равна:

$$R = 0,9999625.$$

Откуда видно, что использование мультиверсионного программного обеспечения на порядок повышает надёжность программной системы.

2. Проблемы создания мультиверсионного программного обеспечения

При создании мультиверсионного программного обеспечения важно учитывать существующие ограничения на время выполнения, потребляемый объём памяти и т.д. Математически это можно записать, как:

$$\sum_{i=1}^n \sum_{j=1}^{m_i} X_{ij} C_{ij} \leq B$$

Где:

C_{ij} – стоимость j -й версии i -го модуля;

B – выделенный бюджет.

Соблюдение указанных условий осложняет труд проектировщика программного обеспечения, повышая длительность разработки и вероятность допущения ошибок при разработке. Для упрощения труда проектировщика предлагается реализация следующего метода формирования мультиверсионного программного обеспечения.

3. Возможный метод формирования мультиверсионного программного обеспечения

Пусть S – допустимая стоимость программной системы, тогда $R(S)$ – вероятность безотказной работы мультиверсионного программного обеспечения, составленного из модулей i ($i = \overline{1 \dots n}$). И $(B - S)$ – возможные затраты, разрешённые для модулей i ($i = \overline{1 \dots n - 1}$).

Рекурсивная формула для вычисления $R_i(S)$ при $i < n$:

$$R_i(S) = \max \left(\left[1 - \prod_{j=1}^{m_i} (1 - R_{ij})^{X_{ij}} \right] * R_{i+1} \left(B - \sum_{j=1}^{m_i} C_{ij} X_{ij} \right) \right),$$

где максимизация происходит по значениям X_{ij} , для которых выполняется:

$$\sum_{j=1}^{m_i} X_{ij} \geq 2,$$

$$\sum_{j=1}^{m_i} C_{ij} X_{ij} \leq S.$$

Для последнего модуля:

$$R_n(S) = \max \left[1 - \prod_{j=1}^{m_n} (1 - R_{nj})^{X_{nj}} \right],$$

где максимизация происходит по значениям X_{nj} , для которых

выполняется:

$$\sum_{j=1}^{m_n} X_{nj} \geq 2,$$

$$\sum_{j=1}^{m_n} C_{nj} X_{nj} \leq S.$$

$R_i(S)$ должны рассчитываться по всем S в последовательности

$$\left[\sum_{k=i}^n \min C_{kj}, \dots, B - \sum_{k=1}^{i-1} \min C_{kj} \right].$$

Заключение

Представленный метод формирования мультиверсионного программного обеспечения позволит успешно решать задачи синтеза мультиверсионного программного обеспечения максимально возможного уровня надёжности с учётом существующих ограничений.