АНАЛИЗ НАДЕЖНОСТИ МУЛЬТИВЕРСИОННЫХ АРХИТЕКТУР АППАРАТНО-ПРОГРАММНЫХ КОМПЛЕКСОВ

Антамошкин О.А., Дегтерев А.С.,
Русаков М.А., Усольцев А.А.
ФГУП ЦКБ «Геофизика»,
Красноярск

На сегодняшний день достаточно сложно с большой точностью оценить надежность какого-либо информационно управляемого аппаратно - программного комплекса. Сбои происходящие в аппаратном обеспечении не могут быть заменены функциями программного обеспечения и наоборот. Анализируя бортовые аппаратно-программные комплексы, следует отметить, что механизмы распространения сбоев оказываются достаточно сложными, а последствия, как правило, трудными или невозможными для прогнозирования [1].

Существует множество моделей оценки как надежности аппаратного, так и надежности программного обеспечения [1-4]. В статье рассматривается одна из моделей, объединяющая в себе мультиверсионную избыточность аппаратной и программной части, используемую для повышения надежности аппаратно-программного комплекса (АПК) в целом.

Иерархическое представление АПК приведено на рисунке 1. Программная система состоит из набора программных модулей. Программные модули выполняют функции посредством выполнения наборов инструкций микропроцессора (или микропроцессоров). Аппаратные компоненты – микропроцессоры, память и другие устройства, непосредственно участвующие в выполнении инструкций.

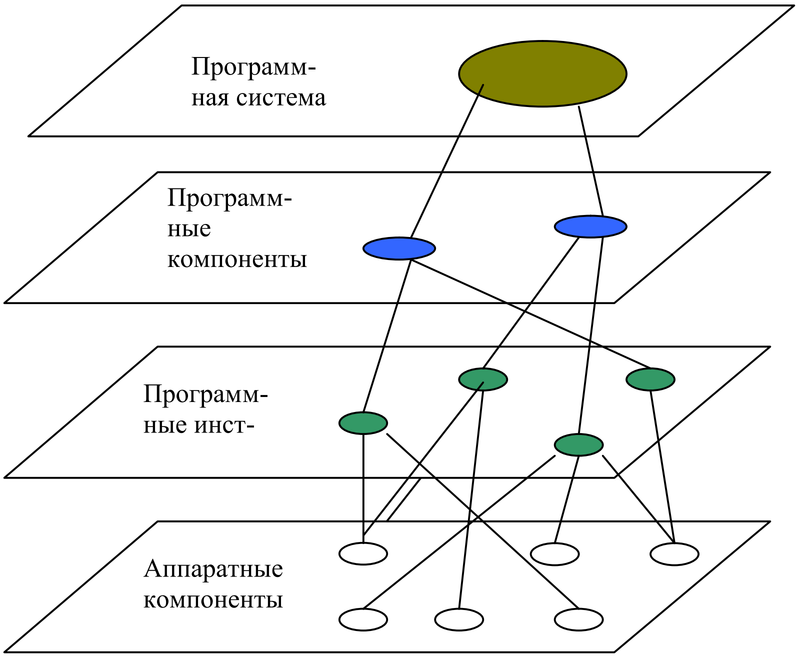

Рисунок 1. Иерархическое представление АПК

Предполагается, что во время простоя системы сбоев не происходит. Кроме того, когда уровни сбоя аппаратных компонент постоянны и в программном обеспечении отсутствуют сбои, то уровень сбоя одной инструкции может быть определен как произведение суммы уровней сбоя аппаратных компонент на время выполнения инструкции [1]:

$$l^{j}_{inst} = m^{j}_{inst} \sum_{i} l^{i}_{hw}\,, \tag{1}$$

где m^{j}_{inst} - время необходимое для выполнения j-й инструкции.

Уровень сбоя в программном модуле можно определить как:

$$l^{j}_{\mathrm{mod}} = \sum_{i} p_i n^{k,i}_{inst}\, l^{i}_{inst}\quad , \tag{2}$$

где p_i - вероятность использования модуля, $n^{k,i}_{inst}$ – общее количество инструкций j в k-м модуле. Здесь p_i определяется операционным профилем архитектуры ПО [2].

Уровень сбоя всей системы определим по формуле:

$$l_{sys} = \sum_{k} l^{k}_{\mathrm{mod}}\ . \tag{3}$$

Как известно ПО, функционирующего без сбоев, практически не бывает. Поэтому формула (3) может быть легко преобразована в выражение, учитывающее сбои в программном обеспечении (без использования отказоустойчивости в аппаратном обеспечении).

$$l_{sys} = \sum_{k} l^{k}_{\mathrm{mod}} + l_{sf}\ . \tag{4}$$

Это значение может быть определено путем тестирования ПО.

Более того, формула (4) может быть расширена до следующего вида:

$$l_{sys} = (1-C)\sum_{k} l^{k}_{\text{mod}} + l_{sf}\,, \qquad (5)$$

где коэффициент C определяется как отношение количества сбоев, устраненных отказоустойчивой системой, к общему количеству сбоев в системе. Данный коэффициент не имеет математического описания и получается опытным путем, например, с использованием имитации сбоев и ошибок в системе [2].

Анализ результатов

В заключение в качестве иллюстрации рассмотрим следующий пример.

Предположим, что в АПК возможно применение аппаратной избыточности (дублирования) и мультиверсионной избыточности ПО.

Исходные данные имеют следующие обозначения:

- количество процессоров: M;
- количество версий ПО: N;
- надежность одного аппаратного модуля: Pi , (i=1,…, M);
- стоимость одного аппаратного модуля: Cpi , (i=1,…, M);
- надежность одной версии ПО: Rj, (j=1,…, N);
- стоимость одной версии ПО: Crj, (j=1,…, N);
- среднее время появления сбоя [3] MTTF= max(MTTFj), (j=1,…, N).

Надежность аппаратно-программного комплекса:

$$W_{sys} = (1-\prod_{i}(1-Pi))(1-\prod_{j}(1-Rj)). \qquad (6)$$

Стоимость аппаратно-программного комплекса:

$$C_{sys} = \sum_{i} Cpi + \sum_{J} Crj\,. \qquad (7)$$

Таблица 1. Пример расчета надежности АПК для разных вариантов архитектур ПО

	Вариант 1	Вариант 2	Вариант 3	Вариант 4
M	1	3	1	3
Pi	0,9	0,9	0,9	0,9
Cpi	500	500	500	500
N	1	1	3	3
Rj	0,8	0,8	0,8	0,8
Crj	200	200	200	200
W	0,720	0,799	0,893	0,991
C	700	1700	1100	2100

Из приведенной таблицы видно, что самый надежный вариант – последний, однако, очевидно, он же обладает и максимальной стоимостью.

СПИСОК ЛИТЕРАТУРЫ

1. Jong Gyun Choi, Hyun Gook Kang. “Reliability Estimation of Nuclear Digital I&C System using Software Functional Block Diagram and Control Flow”. FastAbstract ISSRE Copyright 2000.

2. Telmo Menezes, Diamantino Costa. “On the Extention of Exeption to Support Software Fault Models”. FastAbstract ISSRE Copyright 2000.

3. Ковалев И.В., Юнусов Р.В. Оценка надежности аппаратно-программного информационно-управляющего комплекса. САКС-2002: Тез. докл. Междунар. науч.-практ. конф. (6-7 дек. 2002, Красноярск)/ СибГАУ. Красноярск, 2002. С. 352-353.

4. Ковалев И.В., Алимханов А.М., Юнусов Р.В. Мультиверсионный метод повышения качества программно-информационных технологий для корпоративных структур//Россия в III тысячелетии: Сборник научных трудов по материалам Всероссийской научной конференции/ Изд-во АМБ, Екатеринбург, 2002. С. 171-173.

РОБАСТНАЯ СТАБИЛИЗАЦИЯ ДИСКРЕТНО - ИМПУЛЬСНЫХ СИСТЕМ

Зюзина Н.Ю.

Рассматривается класс систем управления, структура (режим) которых скачкообразно изменяется во времени в соответствии с эволюцией однородной Марковской цепи. В каждом фиксированном состоянии (режиме) объект управления описывается разностным уравнением. В момент скачкообразного изменения режима вектор состояния объекта может изменяться скачком. Число режимов конечно и процесс смены режимов доступен наблюдению. Получено условие стабилизации системы в управлении со статической обратной связью по выходу объекта, переключаемой синхронно со сменой режима, которое стабилизирует систему в случае неопределенности параметров смены режима.

Рассмотрим дискретно – импульсную систему управления, математическая модель которой описывается семейством уравнений

$$\begin{aligned} &\overline{\mathbf{x}_{n+1}} = [\mathbf{A}(r_n)+\mathbf{F}(r_n)\Omega\,(n,r_n)\mathbf{E}(r_n)]\mathbf{x}_n + \mathbf{B}(r_n)\mathbf{u}_n, \\ &\mathbf{y}_n = \mathbf{C}(r_n)\mathbf{x}_n, \\ &\mathbf{x}_{n+1} = \mathbf{\Phi}_{ij}\,\overline{\mathbf{x}_{n+1}}, \end{aligned} \qquad (1)$$

где $\mathbf{x}_n$ – n-мерный вектор состояния объекта; $\mathbf{u}_n$ – k-мерный вектор управления; $\mathbf{y}_n$ – s-мерный вектор выхода объекта; r_n – однородное дискретное состояние цепи Маркова, описывающее процесс смены режима объекта на множестве $N=\{1,2,\mathbf{K},n\}$ и матрицей вероятностей перехода $\mathbf{P}=[P_{ij}]_1^n$ от режима $r_n = i$ до режима $r_{n+1} = j$; $\mathbf{A}_i, \mathbf{B}_i, \mathbf{E}_i, \mathbf{F}_i \quad (i \in \mathrm{N})$ – известные матрицы