

УДК 004.273

А.И. СИРОТЮК

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина***АРХИТЕКТУРИРОВАНИЕ ГАРАНТОСПОСОБНЫХ ВЕБ-ИНФРАСТРУКТУР:
АНАЛИЗ БАЗОВЫХ ВАРИАНТОВ**

Приведены ключевые понятия и определения изучаемой проблематики. Изложены основные требования к инфраструктурам при проектировании и реализации «успешных» проектов. Предложены и рассмотрены варианты архитектур гарантоспособной веб-инфраструктуры в области медицины. Проведен первый этап формализации выделенных параметров системы. Сформулировано дальнейшее направление исследований.

Ключевые слова: инфраструктура, архитектурирование, сервис-ориентированная архитектура, гарантоспособность.

Введение

В настоящее время для ряда приложений растут объемы информации, которую большинство систем, ввиду тех или иных причин, уже не в состоянии обрабатывать в полной мере. Проблемными аспектами являются обеспечение надежности и актуальности обмениваемых данных и системы в целом [1 – 5]. Существует большое количество областей, в которых информационно-аналитические и управляющие компьютерные системы объединяются в инфраструктуры. Тогда вышеперечисленные проблемы прогрессируют в большей степени, и управление такого рода инфраструктурами становится затруднительным, не охватывающим все стороны корректной работы с данными, а иногда попросту невозможным, так как архитектуры и модели, разработанные для ранее используемых систем, могут не соответствовать обрзовавшимся конгломератам.

Таким образом, актуальными являются разработки в области архитектурирования, моделирования и исследования гарантоспособных инфраструктур.

**1. Основные требования
к «успешным» проектам**

Дадим определение некоторым ключевым понятиям в данной области.

Инфраструктура – система организационных структур, обеспечивающих функционирование и развитие информационного пространства страны и средств информационного взаимодействия. Информационная инфраструктура включает совокупность информационных центров, банков данных и знаний, систем связи обеспечивает доступ потребителей к информационным ресурсам.

Архитектура – сочетание схем организации, предметизации и навигации, реализованных в информационном пространстве. **IT-архитектура** – структура и организация информационных процессов.

Сервис-ориентированная архитектура (СОА) – это компонентная модель, которая связывает различные функциональные модули приложений, называемые веб-сервисами (или службами), посредством четко определенных интерфейсов и соглашений между этими сервисами. Интерфейсы определяются независимым способом и не должны зависеть от аппаратной платформы, операционной системы или языка программирования, на котором реализован сервис.

Веб-сервис – это программный интерфейс, который описывающий набор операций, которые могут быть вызваны удаленно по сети посредством стандартизированных XML сообщений. Для описания вызываемой операции или данных используются протоколы, базирующиеся на языке XML.

Актуальность информации – свойство данных в указанный (по умолчанию – текущий) момент времени адекватно отображать состояние объектов предметной области.

Далее следуют общие требования к системам (в частности инфраструктурам), которых необходимо придерживаться при проектировании и реализации «успешных» проектов.

Открытость – система должна быть построена на открытых стандартах с использованием стандартизированных протоколов.

Масштабируемость – система должна быть легко масштабируемой по способам использования, размеру, географии.

Прозрачность – для клиентов системы (человека или внешних систем) вся система должна выглядеть единой информационной системой, в

которой расположение ресурсов не имеет значения и скрывается.

Переносимость – компоненты системы не должны зависеть от программной (операционной системы) и аппаратной платформы узлов.

Гарантоспособность – это способность системы предоставлять требуемые услуги, которым можно оправданно доверять.

Надежность – свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения. Надежность является комплексным свойством, которое в зависимости от назначения объекта и условий его применения может включать безотказность, долговечность, ремонтпригодность и сохраняемость или определенные сочетания этих свойств [6].

Устойчивость – система должна быть стойкой к внешним воздействиям, в том числе стихийным бедствиям и к внутренним сбоям.

Безопасность – в системе должны быть реализованы все механизмы безопасности информации, разграничения доступа, авторизации и т.д.

Доступность – это уровень обслуживания, предоставляемый приложениями, службами или системами. Также трактуется как состояние информации, при котором субъекты, имеющие право доступа (право на чтение, изменение, копирование, уничтожение информации, а также права на изменение, использование, уничтожение ресурсов), могут реализовывать их беспрепятственно.

Для повышения уровня доступности необходимо внедрить механизмы повышения отказоустойчивости для маскировки или снижения влияния сбоев компонентов и зависимостей службы. Повышение отказоустойчивости достигается путем внедрения избыточности для подверженных сбоям компонентов без дублирования.

Интероперабельность – система должна быть способна взаимодействовать с другими (внешними) системами.

Эволюционность – система должна быть способной на всем протяжении жизненного цикла к эволюционному развитию и совершенствованию.

Поддержание логической целостности данных – все виды взаимодействия, как внутри системы, так и с внешними системами должны реализовываться в виде транзакций.

Катастрофоустойчивость – это возможность восстановить работоспособность приложений обеспечивающих непрерывность бизнеса в течение приемлемого периода времени. Катастрофой можно считать любое событие, ведущее к остановке сервиса, обеспечивающего непрерывность бизнеса, или

потере данных. Катастрофоустойчивая система представляет собой дальнейшее развитие систем высокой доступности.

Данные требования должны быть соблюдены на каждой стадии жизненного цикла проекта. Особое внимание необходимо уделить проектированию системы, формированию ее архитектуры и выбору технологий.

В результате тщательного ознакомления с выбранной сферой деятельности и требованиями заказчика, формируется спецификация и техническое задание на проект. В соответствии с данными документами уже можно определить назначение системы и ее основные функции.

2. Выбор архитектур системы

Проанализировав ряд существующих информационно-аналитических и управляющих систем в основном в области медицины [1– 5, 8], были выделены следующие функции систем обмена, обработки и хранения значительных объемов информации в инфраструктурах:

- надежное хранение информации;
- обеспечение доступности и безопасности информации;
- актуальность и достоверность информации;
- автоматизация документооборота;
- автоматизация бизнес-процессов;
- автоматизация административных бизнес-процессов и логистик;
- наличие аналитической системы для обработки и анализа данных;
- автоматизация сбора информации от оборудования (если необходимо);
- интеграция с внешними системами.

На основании данных функций необходимо определить программную и системную архитектуры системы [7].

Существует несколько вариантов программной архитектуры:

- двухзвенное приложение клиент-сервер;
- Enterprise Architecture (EA);
- сервис-ориентированная (SOA);
- сервис-ориентированная с интеграцией сервисов в единое информационное пространство.

Предлагаемым решением проблемы выбора программной архитектуры для инфраструктур и сложных систем является сервис-ориентированная архитектура с интеграцией сервисов в единое информационное пространство.

Существуют следующие виды системных архитектур:

- централизованная;
- федеративная;

- полносвязная (“peer-to-peer”);
- централизованная на основе кластера;
- иерархическая на основе геокластера.

Последний вариант имеет наибольшее число преимуществ для рассматриваемых инфраструктур [8]. Именно данная архитектура была выбрана для проектирования инфраструктуры в сфере здравоохранения. Данную архитектуру можно также использовать для проектов в других областях.

Большим преимуществом геокластера является удовлетворение нескольких требований по гаранто-способности инфраструктуры за счет гибкости, а также внутренних ресурсов технологии. Рассмотрим некоторые из них [9].

Катастрофоустойчивость обеспечивается применением технологии геокластера на национальном уровне (например, Sun Geo Cluster).

Безотказность обеспечивается:

- применением технологий HA-кластеров на всех уровнях (например, на основе Open HA Cluster).
- введением избыточности каналов передачи информации между узлами системы;
- введением избыточности точек доступа в систему;
- введением избыточности и диверсности сервисов в системе;
- использованием программной и аппаратной диверсности.

Безопасность обеспечивается:

- использованием безопасных протоколов обмена информацией https, ssl, ipsec;
- применением современных технологий обеспечения безопасности информации с постоянным обновлением требований и реинжинирингом подсистем информационной безопасности;
- ограничением перечня узлов, которым разрешен доступ к геокластеру.

Выбор программной и системной архитектуры инфраструктуры является не последним шагом в построении самой архитектуры проекта. Данную архитектуру можно представить в виде нескольких вариантов. Далее рассматривается простейший пример и анализируется архитектура, состоящая из двух звеньев [10].

Классификационными признаками системы, в соответствии с которыми будут различаться варианты архитектуры, являются:

- географическое разделение серверов;
- характер клиентов;
- активный зеркальный сервер;
- наличие промежуточной среды (middleware).

Всего рассматривается 4 варианта архитектур. Каждый вариант имеет две серверные части, два

клиента, потоки заявок от первого клиента, потоки заявок от второго клиента, поток обмениваемых данных между серверами, который также разбивается на два потока.

Система S обладает рядом выделенных параметров, что можно формально описать следующим образом:

$$S = \{k, c, \lambda, A, V, t_c\}, \quad (1)$$

где $k = \{k_1, k_2\}$ – общий поток заявок клиентов,

k – поток заявок клиента 1 и клиента 2,

$c = \{c_{12}, c_{21}\}$ – общий поток заявок между серверами,

c – потоки заявок от C_1 к C_2 и от C_2 к C_1 ,

C_1 – сервер 1,

C_2 – сервер 2,

c_{12} – потоки заявок от C_1 к C_2 ,

c_{21} – потоки заявок от C_1 к C_2 и от C_2 к C_1 ,

A – актуальность данных,

V – объем обмениваемых данных между серверами,

t_c – время обмена данными между серверами.

Далее рассмотрим каждый вариант архитектуры более подробно.

В первом варианте архитектуры (рис. 1) заявки клиентов поступают только на активный сервер C_1 . Второй сервер C_2 является полным зеркалом первого, т.е. через установленное время t , информация, поступающая на сервер C_1 , полностью копируется на сервер C_2 .

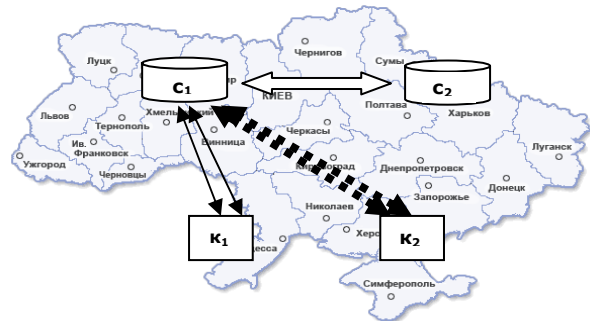


Рис. 1. Первый вариант архитектуры с одним активным сервером

Особенностью второго варианта архитектуры (рис. 2) является географическое разделение заявок клиентов. Заявки от клиента k_1 поступают только на сервер C_1 , а заявки от клиента k_2 поступают только на сервер C_2 . Таким образом, разгружается трафик, т.е. интенсивность заявок уменьшается, но возникает вопрос актуализации данных.

Третий вариант архитектуры (рис. 3) отличается наличием промежуточного устройства (middleware). Заявки от всех клиентов сначала поступают на middleware, которое определяет, какой из серверов менее нагружен и отправляет на него заявки. В таком случае разгружается очередь заявок от всех клиентов, однако появляется дополнительный компонент ненадежности.

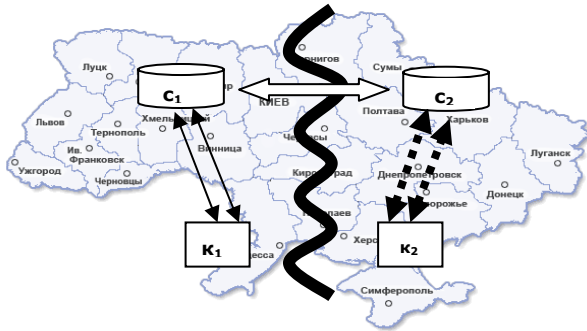


Рис. 2. Второй вариант архитектуры с географическим разделением заявок клиентов

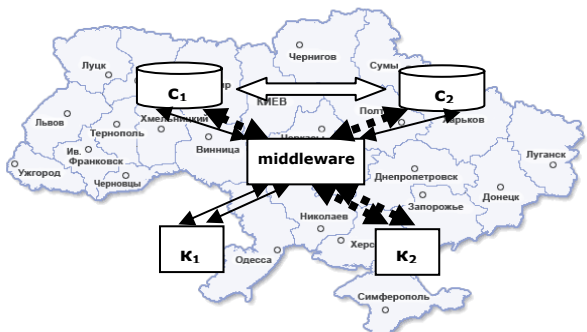


Рис 3 Третий вариант архитектуры с middleware

В четвертом варианте архитектуры (рис. 4) заявки от каждого клиента поступают на каждый сервер. Таким образом, уменьшаются до минимума потери заявок и объем данных, обмениваемых между серверами. Однако интенсивность заявок и нагрузка на каждом сервере увеличивается.

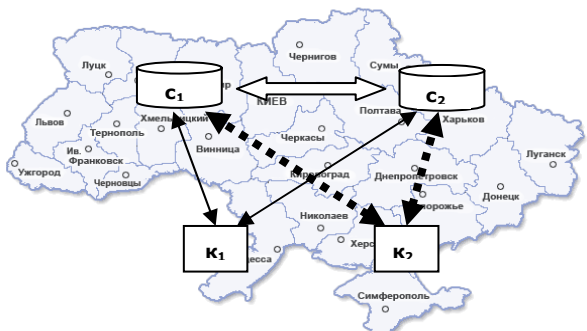


Рис. 4 Четвертый вариант архитектуры с заявками на все сервера

Указанные в (1) параметры для вышеописанных архитектур были сведены в общую таблицу (табл. 1). Данная таблица является первым шагом формализации потоков заявок между клиентами и серверами инфраструктуры, а также потоков заявок между серверами.

Здесь необходимо уточнить обозначения некоторых символов, которые не были описаны ранее или могут иметь двойной смысл. Таким образом:

$k \Rightarrow C_1$ – поток заявок k поступает на сервер 1,
 $C_1 | C_2$ – сервер 1 или сервер 2,

$k_1 + k_2$ – сумма потоков заявок от клиента 1 и клиента 2,

$c_{12} \rightarrow \min$ – поток заявок от первого сервера ко второму стремится к минимуму,

$c_{12} = k_1(t)$ – поток заявок от сервера 1 к серверу 2 равен (не учитывая потери) потоку заявок от клиента 1 за определенное время t .

↑ – значение параметра высокое,

↑↑ – значение параметра значительно повышается,

↓ – значение параметра невысокое,

↓↓ – значение параметра значительно понижается.

Таблица 1

Сводная таблица основных параметров системы

Вар. арх-ры	k	c	λ	A	V
1	$k_1 + k_2 \Rightarrow C_1$	$c_{12} = k_1(t) + k_2(t)$ $c_{21} \rightarrow \min$	λ ↑↑	A ↑	V ↑↑
2	$K / 2 \Rightarrow C_1, C_2$ $k_1 \Rightarrow c_1$ $k_2 \Rightarrow c_2$	$c_{12} = k_1(t)$ $c_{21} = k_2(t)$	λ ↓	A ↓	V ↑
3	$K / 2 \Rightarrow C_1, C_2$ $k_1 \Rightarrow C_1 C_2$ $k_2 \Rightarrow C_1 C_2$	$c_{12} = k_{m2}$ $c_{21} = k_{m1}$	λ ↓↓	A ↓	V ↑
4	$k_1 + k_2 \Rightarrow C_1$ $k_1 + k_2 \Rightarrow C_2$	$c_{12} \rightarrow \min$ $c_{21} \rightarrow \min$	λ ↑↑	A ↑↑	V ↓↓

Заключение

Сформированные варианты архитектур требуют дальнейшей проработки IT специалистов, которые будут определять, какими технологиями воспользоваться при дальнейшей разработке проекта. Данная миссия является достаточно сложной, так как каждый день появляются новые технологии, которые могут быть действительно полезными, но могут быть и «засорением» IT рынка. К тому же новейшие технологии далеко не всегда являются стандартизированными в той или иной области, что ухудшает открытость и гибкость систем, использующих их. С другой стороны, существует опасность старения или даже отмирания устоявшихся технологий. Поэтому необходимо тщательный анализ как стандартных технологий в конкретной области и в мире, так и специализированных технологий (например, HL7 в медицине) и целесообразность их применения в данном проекте.

Подбор технологий должен быть произведен с учетом выбранной сервис-ориентированной архитектуры с интеграцией сервисов в единое информационное пространство, а также иерархической архитектуры инфраструктуры на основе геокластера, а также учитывая математическую модель данной инфраструкту-

ры. Поэтому, следующим этапом исследования данной проблематики является построение моделей СМО (система массового обслуживания), тщательный анализ математических моделей, для соответствующих выбранным вариантам архитектур.

Литература

1. *A computer scientist's reactions to NPfIT [Электронный ресурс] / B. Randell // Journal of IT Ness Technologies. – 2007. – P. 1-13. – Режим доступа: www.ness.com.*

2. Syrotyuk A. *EPR systems analysis. [Электронный ресурс]. – Режим доступа: http://dependability.org.ua/downloads/reports/epr_analysis.ppt.*

3. ProClarity Corporation, 2003 [Электронный ресурс]. – Режим доступа: www.proclarity.com

4. Cimino J. *Twenty Years of IAIMS: The Columbia University New York Presbyterian Hospital Clinical Data Repository / J. Cimino // IAIMS Consortium Annual Meeting. – 2005. – P. 1-44.*

5. Bolgov M. *Medical establishments automation / M. Bolgov – K.: Publ. Kupriyatnova, 2006. – 494 p.*

6. ГОСТ 27.002-89. – *Надежность в технике. Основные понятия определения. – Введ. 01.07.90. – М.: Изд-во стандартов, 1990. – 39 с.*

7. Antonyan I.M. *WEB-system of automatic forming and analysis electronic case history / I.M. Antonyan, V.M. Lesovoy, K.O. Bokhan, O.O. Gordeev, Y.I. Nezhuta, V.S. Kharchenko, S.V. Kharchenko, A.V. Shus // Abstracts of Anniversary Conference of Urologists. – Kharkiv: KhMAPE, 2007. – P. 147-149.*

8. Катастрофоустойчивые геокластерные системы Проминвестбанка и АКБ «Укрсоцбанк» [Электронный ресурс]. – Режим доступа: <http://www.it.sitronics.com/sub/it/ru/projects/200512201205-9939.htm>.

9. Syrotyuk A. *Medical information systems: analysis of development and application experience / A. Syrotyuk, K. Bokhan, O. Oksuchenko, V. Kharchenko // Радіоелектронні і комп'ютерні системи. – Харків: НАКУ «ХАІ». – 2008. – № 7 (33). – С. 32-38.*

10. Сиротюк А.И. *Анализ и разработка вариантов архитектур медицинских информационных систем / А.И. Сиротюк // Міжнародна науково-технічна конференція ІКТМ-2008: Тези доповідей. Т. 2. – X.: НАКУ "ХАІ". - 2008. – С. 202.*

Поступила в редакцію 20.02.2009

Рецензент: д-р техн. наук, проф., зав. кафедри А.Л. Ляхов, Полтавський національний технічний університет, Полтава.

АРХИТЕКТУРУВАННЯ ГАРАНТОЗДАТНИХ ВЕБ-ІНФРАСТРУКТУР: АНАЛІЗ БАЗОВИХ ВАРИАНТОВ

А.І. Сиротюк

Приведені ключові поняття та визначення проблематики, що вивчається. Викладені головні вимоги до інфраструктур, яких необхідно дотримуватися під час проектування та реалізації «успішних проєктів». Запропоновані та розглянуті варіанти архитектур гарантоздатної веб-інфраструктури в області медицини. Проведено перший етап формалізації виділених параметрів системи. Сформульовано подальший напрямок досліджень.

Ключові слова: інфраструктура, архітектуровання, гарантоздатність.

DEPENDABLE WEB-INFRASTRUCTURE ARCHITECTURING: ANALYSIS OF BASE VARIANTS

A.I. Syrotyuk

Main notions of studied agenda are given. Main requirements to infrastructures are stated that should be held to, while successful projects designing and implementation. Dependable web-infrastructure architecture variants in the area of medicine are suggested and considered. First stage of singled out system parameters formalization is held. Further research approaches are formulated.

Key words: infrastructure, architecturing, dependability.

Сиротюк Анна Ивановна – аспірант, асистент кафедри комп'ютерних систем і мереж Національного аерокосмічного університету ім. Н.Е. Жуковського «ХАІ», Харків, Україна, e-mail: annasyrotyuk@yandex.ru.