

## **Методика проектирования баз данных для автоматизированных систем управления специального назначения**

**Дудаков Н.С.<sup>1\*</sup>, Макаров К.В.<sup>1\*\*</sup>, Тимошенко А.В.<sup>2\*\*\*</sup>**

<sup>1</sup>*Группа компаний «РТИ», ул. 8 Марта, 10, стр. 1, Москва, 127083, Россия*

<sup>2</sup>*Радиотехнический институт имени академика А.Л. Минца, ул. 8 Марта, 10, стр. 1, Москва, 127083, Россия*

*\*e-mail: nikolay.dudakov@gmail.com*

*\*\*e-mail: kmakarov@oaorti.ru*

*\*\*\*e-mail: atimoshenko@rti-mints.ru*

### **Аннотация**

Предложена методика формализации требований к системам управления базами данных (СУБД) при проектировании АСУ специального назначения. Показано, что при решении разнородных высокодинамичных задач применение универсальных СУБД неэффективно, что обуславливает актуальность разработки специальной СУБД с учетом поставленных требований. Проведена декомпозиция задач АСУ Воздушно-Космическими силами (ВКС), предложено использование распределенной неоднородной СУБД.

**Ключевые слова:** системы управления базами данных, автоматизированные системы управления, требования к программному обеспечению

## **Введение**

В процессе развития автоматизированных систем управления (АСУ) специального назначения, темпы обмена данным и количество передаваемой информации постоянно возрастают, при этом, требования по времени цикла боевого управления ужесточаются, в связи с чем возрастает важность задачи проектирования специализированных средств хранения информации. Для устойчивой, надежной и производительной работы АСУ в качестве механизмов хранения используются системы управления базами данных (СУБД). СУБД является централизующим звеном в работе АСУ, от характеристик её работы, от удобства доступа к данным зависит работа всего комплекса вооружений, при большом количестве обращений к СУБД она может быть узким местом комплекса с точки зрения производительности. Так, одной из задач, стоящих при разработке АСУ Вооруженных сил 2008-2012 было снижение с 50% до 5% времени цикла боевого управления, затрачиваемого на сбор и обработку данных об обстановке [1].

### **Постановка задачи проектирования СУБД АСУ ВКС**

В работе рассматриваются вопросы проектирования СУБД для АСУ стратегического звена управления Воздушно-Космическими силами (ВКС), позволяющей эффективно, с достаточной производительностью и удобством в использовании выполнять задачи АСУ в части Военно-воздушных сил (ВВС) и Противовоздушной обороны (ПВО).

Проектирование и исследование баз данных представляет собой динамично развивающуюся область разработки программного обеспечения. Вопросы разработки и проектирования СУБД широко представлены в научной и прикладной литературе, как в классических работах (К. Дж. Дейт, Э. Кодд, Т. Коннелли), так и в материалах по современным реализациям СУБД (*Oracle, DB2, PostgreSQL*, Линтер) [2].

Среди научных подходов по проектированию, анализу и систематизации СУБД, опубликованных в российских изданиях, можно выделить работы Кузнецова С.Д., Плутенко А.Д., Харитоновой И.А., Тихомирова Ю.В., Григорьева Ю.А, и других.

Методика проектирования современных СУБД предусматривает три уровня проектирования – концептуальное (конструирование информационной модели), логическое (выбор и применение модели данных к информационным сущностям) и физическое (реализация логической структуры данных в целевой СУБД) [3].

Представленная в научной литературе методика имеет ряд недостатков:

1. отсутствует научно-методический аппарат по выбору целевой СУБД, что приводит к повсеместному применению наиболее универсальных программных решений;
2. отсутствует формализация обратных связей при проектировании, что затрудняет коррекцию логической модели при проектировании программных решений.

В большинстве современных проектов в качестве универсальной СУБД используется клиент-серверная объектно-реляционная промышленная СУБД *PostgreSQL*. Зачастую, данная СУБД удовлетворяет поставленным при проектировании требованиям, тем не менее, при высокой нагрузке, время выполнения ряда запросов может быть уменьшено на 90% заменой промышленной СУБД на более простую и, в ряде случаев, быструю локальную БД. Большое число запросов на чтение является узким местом современных клиент-серверных СУБД [4].

Для сравнения производительности баз данных при определенной нагрузке построена зависимость времени выполнения запросов на чтение для промышленной клиент-серверной СУБД, сервер которой расположен локально и удаленно соответственно, а также для движка БД *SQLite* – простого и производительного локального хранилища данных (Рис.1):

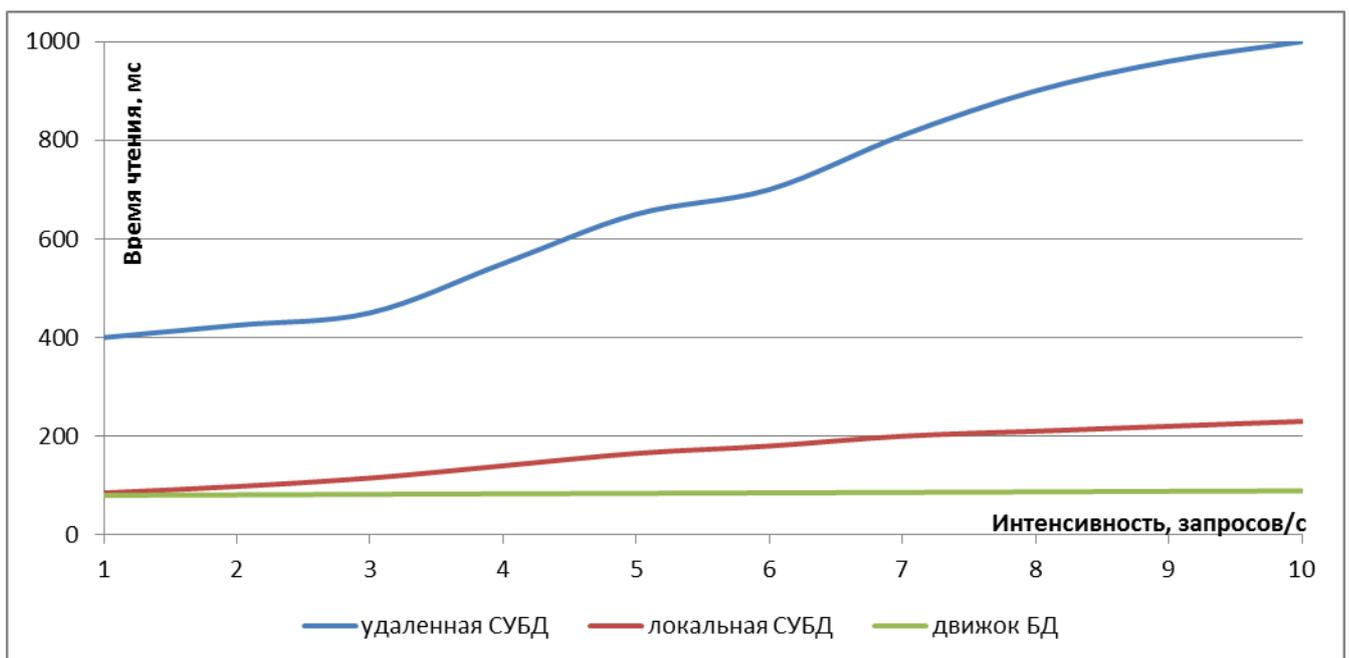


Рис.1 Время чтения данных при повышении интенсивности запросов

Построенная зависимость времени чтения данных при росте интенсивности запросов показывает существенную зависимость времени чтения данных от состояния и загрузки каналов между кластером серверов и АРМ. Также, использование более простой и быстрой локальной БД позволяет сохранить малое время чтения данных при значительном росте нагрузки. В то же время, возможностей простых и быстрых локальных БД не достаточно для выполнения всех функциональных требований.

Таким образом, отсутствие универсальной СУБД достаточной эффективности, ряд недостатков классической методики проектирования СУБД обосновывают разработку специального методического аппарата, основанного на логической формализации, как состава и структуры СУБД, так и собственно процесса его проектирования.

### **Методика формализации требований к СУБД**

Возможным подходом к проектированию СУБД является:

1. На этапе концептуального проектирования СУБД сформировать пространство требований к СУБД ***P***, включающее в себя:
  - а. требования выполнения: применимость СУБД к целевой платформе и системам специального назначения;
  - б. функциональные требования: наличие механизмов доступа к данным, резервирования, транзакционность изменений, и т.д.;

в. динамические требования: поддержка заданного количества сущностей с известными темпами запросов на обновление и доступ к данным.

2. Осуществить моделирование задач  $\mathbb{Z}_t \in \mathbb{T}\mathbb{Z}$  пространства задач  $\mathbb{T}$ , как клиентов СУБД, определив информационные сущности и сформировав область пространства  $\mathbb{P}_{\mathbb{T}\mathbb{Z}}$  требований к СУБД комплекса.

3. Решать задачу проектирования СУБД, как задачу оптимизации (минимизации): ограничениями являются условие покрытия целевой СУБД  $D_i$  (набором СУБД  $\bigcup_{D \in \mathbb{D}} D_i$  в общем случае) области требований  $\mathbb{P}_{\mathbb{T}\mathbb{Z}}$  и рациональные затраты  $S(D_i)$  на разработку и эксплуатацию СУБД, критерием – среднее время выполнения запросов. Формализуя, в приведенных обозначениях, по набору входных задач требуется найти СУБД (набор СУБД), такой, что:

$$t_{\text{СУБД}} \left( \bigcup_{\mathbb{Z}_t \in \mathbb{T}\mathbb{Z}} \mathbb{Z}_t, \bigcup_{D \in \mathbb{D}} D_i \right) \rightarrow \min, \quad (1)$$

при ограничениях:

$$\bigcup_{t=1, \overline{T}} \mathbb{P}_t = \bigcup_{\mathbb{Z}_t \in \mathbb{T}\mathbb{Z}} M(\mathbb{Z}_t) \subseteq \bigcup_{D \in \mathbb{D}} P(D_i), \quad (2)$$

$$S \left( \bigcup_{D \in \mathbb{D}} D_i \right) \leq S_{\max}, \quad (3)$$

где  $\mathbb{D}$  – множество рассматриваемых целевых СУБД;

$M: \mathbb{T} \rightarrow \mathbb{P}$  – преобразование, ставящее в соответствие задачам  $\mathbb{T}\mathbb{Z}$  вектор требований пространства  $\mathbb{P}$ ;

$P: \mathbb{D} \rightarrow P$  – преобразование, ставящее в соответствие целевым СУБД область пространства требований  $P$ .

Увеличение количества СУБД набора  $D$  (по классификации Дейта – распределенная неоднородная СУБД, РН СУБД [5]) уменьшает доступность СУБД при запросах одновременно к двум и более СУБД. Соответственно, возможность применения неоднородной СУБД должна быть обусловлена, как минимум, наличием в общем наборе требований  $P_{T3} = \bigcup_{t=1}^T p_t$  слабо связанных областей пространства  $P$ , позволяющих провести кластеризацию информационных сущностей, таким образом, что количество запросов, требующих данные из нескольких кластеров одновременно, было минимально. При этом, потенциальное преимущество использования нескольких СУБД тем больше, чем более разнородным является набор  $P_{T3}$ . Соответственно, применение РН СУБД является актуальным при возможности кластеризации набора  $P_{T3}$  на части  $P_i: P_{T3} = \bigcup_{i=1}^M P_i$ , таким образом, что расстояние между кластерами велико:

$$dist_{p,q=\overline{1..M}, p < q}(P_p, P_q) \gg dim(P_p) + dim(P_q), \quad (4)$$

а количество пересекающихся требований - минимально,

$$\bigcap_{i=1}^M P_i \rightarrow \emptyset. \quad (5)$$

Схема декомпозиции исходного набора задач представлена на рисунке 2:

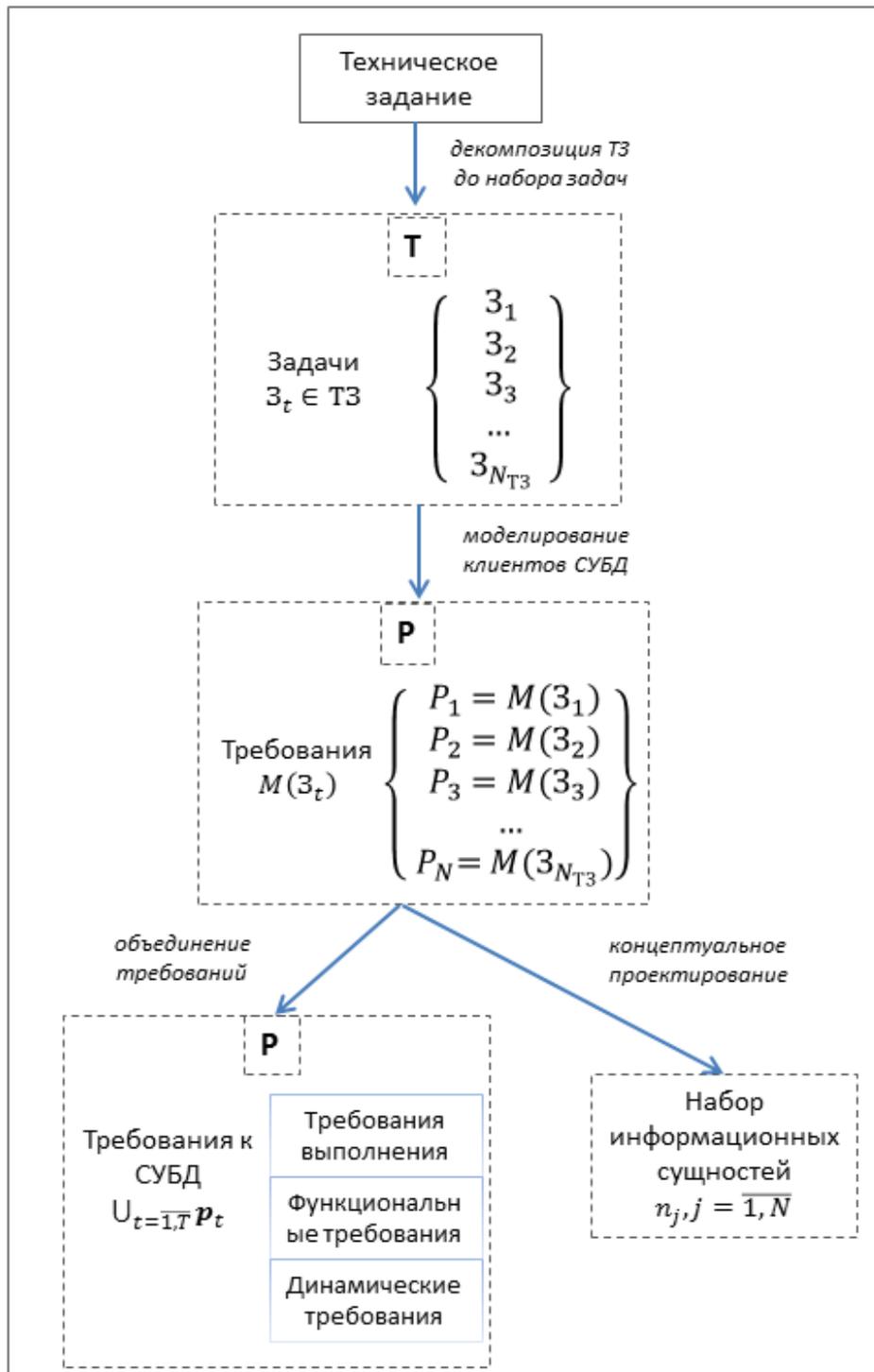


Рис.2 Методика формирования требований к СУБД

Представленная методика формализует процесс поиска целевой СУБД на этапе концептуального проектирования. На основании ТЗ формализуются требования каждой задачи, как клиента СУБД, на основании которых, в результате композиции, строится пространство требований к СУБД комплекса.

## Декомпозиция задачи проектирования АСУ ВКС

Анализ динамических требований при разработке АСУ ВКС стратегического звена управления показывает, что они могут быть разделены на две области, соответственно обрабатываемой информации, по темпам запросов, на «статическую» и «динамическую» части. Статические данные характеризуются низкой частотой обновления и доступа, к ним можно отнести классификационные данные, различную справочную информацию. Статическая информация имеет достаточно большой объем, и целесообразно хранить её на серверах баз данных. Динамические данные характеризуются высоким темпом обновления, а также необходимостью отслеживания изменений в режиме реального времени. Динамическая часть имеет сравнительно меньший объем данных и целесообразно хранить её как на серверах, так и на локальных рабочих местах для повышения производительности чтения. Выделение третьей обособленной группы данных не представляется целесообразным, т.к. увеличивает количество связей между хранилищами за счет разнесения связанных классов данных между частями системы, что негативно сказывается на производительности и надежности комплекса. Разбиение данных с учетом «времени жизни» объектов оставляет только однонаправленные связи классов, т.е., ссылки из динамической части в статическую.

Учитывая также практику утверждения протоколов информационно-логического, технического взаимодействия (ПИЛТВ) и постановок задач, динамические требования к системе можно считать постоянными. Применение РН СУБД в данном случае позволяет провести оптимизацию структуры СУБД за счет

эффективного распределения нагрузки по частям СУБД при постоянном разбиении множества информационных сущностей по частям РН СУБД  $D_i$ .

Рассматриваемая в статье задача проектирования СУБД была решена с использованием клиент-серверной СУБД *PostgreSQL* из состава *Astra Linux* и распределенной реляционной СУБД с полной репликацией на основе движка БД *SQLite* (Динамическое Хранилище Данных, ДХД [6]) для статической и динамической части данных соответственно (Рис.3):

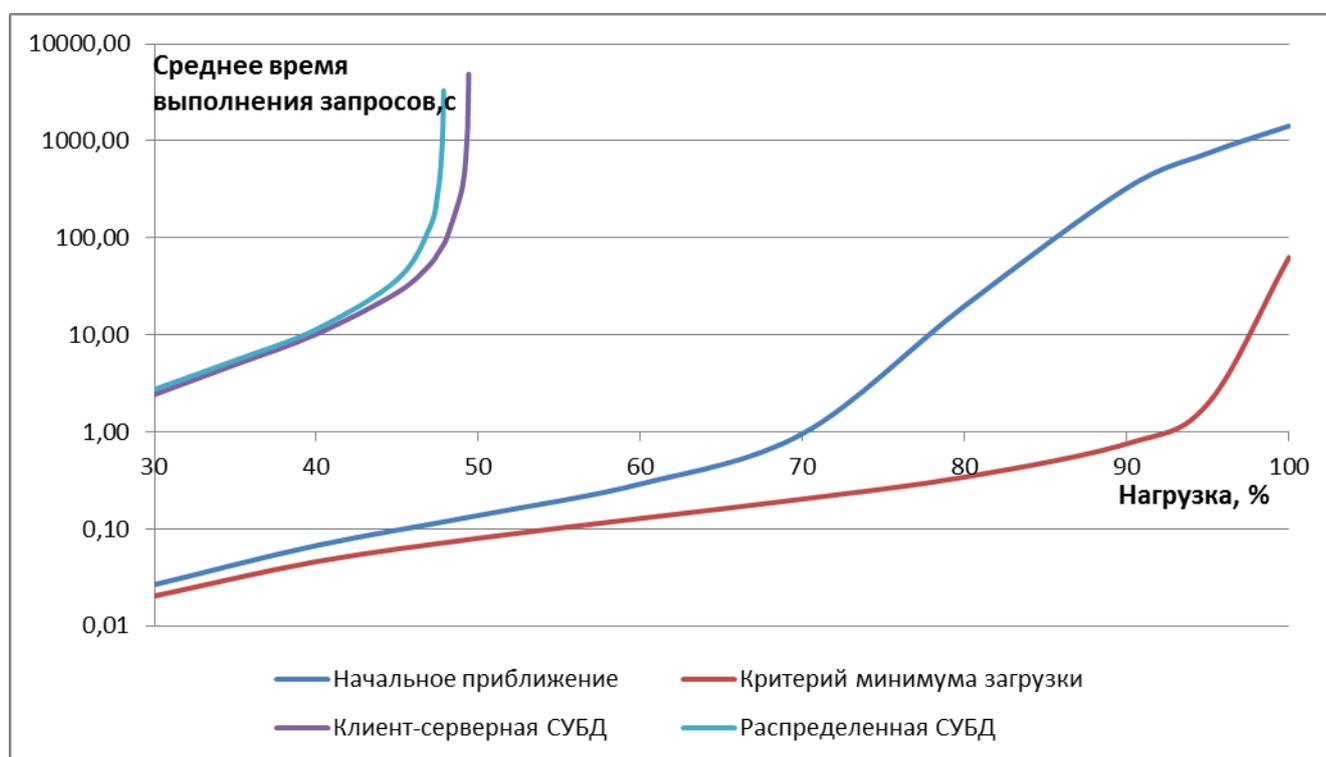


Рис.3 Время выполнения запросов при изменении нагрузки

Решение задачи (Рис.3) показывает эффективность применения РН СУБД при обработке практических данных при выполнении условий (4),(5). Применение РН СУБД позволяет увеличить нагрузку на СУБД более чем в два раза. Кроме того, управление структурой СУБД по принципу минимума нагрузки (разбиение при

обработке класса определяется минимумом времени обработки в незагруженной СУБД) также значительно позволяет улучшить производительность обработки запросов. Построение разбиения по данному принципу является задачей оптимизации и может быть осуществлено с помощью методов линейного программирования [7].

### **Заключение**

1. Предложена методика формирования и анализа требований к СУБД при проектировании комплекса вооружений, согласно которой, при наличии разнородных малосвязанных данных, предлагается использование неоднородной распределенной СУБД, состоящей из нескольких различных программных решений. Показано, что при фиксированных на этапе проектирования требованиях к СУБД, эффективно постоянное разбиение хранимых данных по частям неоднородной СУБД, позволяющее дополнительно управлять структурой СУБД за счет построения соответствия частей данных и СУБД, обеспечивающего лучшую производительность.

2. Проведена декомпозиция задач АСУ ВКС, показано, что специфика хранимых данных - объем и разнородность информации, темпы запросов, обуславливают требования к СУБД, не позволяющие при автоматизации стратегического звена управления ВКС использовать одно готовое открытое программное решение. На основании разработанной методики, предложен и обоснован подход, согласно которому при разработке СУБД АСУ ВН предлагается использование распределенной неоднородной СУБД из двух частей – для

«статических» и «динамических» данных соответственно, что вдвое увеличивает возможности СУБД по обработке данных.

*Работа выполнена при поддержке гранта Президента РФ*

*(№ НШ-6831.2016.8).*

### **Библиографический список**

1. Андреев В.Н. 5 этапов развития АСУ // Военно-космическая оборона. 2011. №2 (57). С. 14-19.
2. Коннолли Т., Бегг М. Базы данных, проектирование, реализация и сопровождение. - М.: Издательский дом «Вильямс», 2003. – 1440 с.
3. Кузнецов С.Д. Основы современных баз данных. – М: Издательство «Интернет-университет информационных технологий», 2007. - 488 с.
4. Брехов О.М., Тин М.А. Обработка запросов к базе данных посредством ассоциативной вычислительной системы // Труды МАИ, 2015, № 84: <http://www.mai.ru/science/trudy/published.php?ID=63152>
5. Дейт К.Дж. Введение в системы баз данных. - М.: Издательский дом «Вильямс», 2001. – 1328 с.
6. Дудаков Н.С., Пирогов Н.Е., Шумилов Ю.Ю. Гибридная система управления хранением данных // Вестник национального исследовательского ядерного университета МИФИ. 2012. Т.1. №1. С. 119-124.
7. Кокуев А.А., Ктитров С.В. Построение дерева решений в задачах

линейного программирования // Вестник национального исследовательского  
ядерного университета МИФИ. 2014. Т.3. №1. С. 119-124.