



**Computer  
Technology  
Innovations  
2020**

# **УПРАВЛЕНИЕ МНОГОУРОВНЕВЫМ ХРАНИЛИЩЕМ ДАННЫХ**



**ГУАП**

**Пойманова Екатерина Дмитриевна,**



**Татарникова Татьяна Михайловна**

**Санкт-Петербург 2020**

# ВВЕДЕНИЕ

- Сегодня имеется широкий ряд разнообразных носителей данных и СХД
- Наблюдается активный рост количества разнородной информации
- На сегодняшний день в процессе хранения не учитывается взаимосвязь характеристик сохраняемых данных с существующими технологиями хранения

# СТРУКТУРА СХД

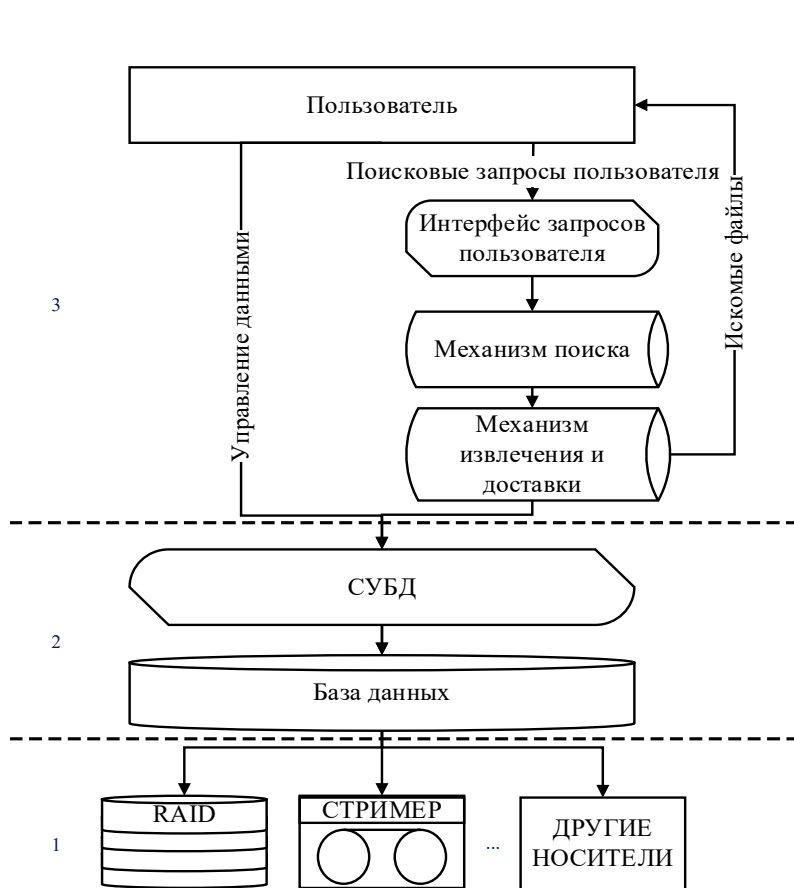


Рисунок 1 – Иерархическая структура СХД



Рисунок 2 – Матричная структура хранилища данных

Хранилище данных было формализовано в виде математической матрицы, где каждая ячейка матрицы – это том или носитель соответствующего уровня хранилища.

Каждая ячейка предназначена для хранения файлов с определенными метрическими характеристиками, которые берутся из метаданных.

# СХД КАК СИСТЕМА УПРАВЛЕНИЯ

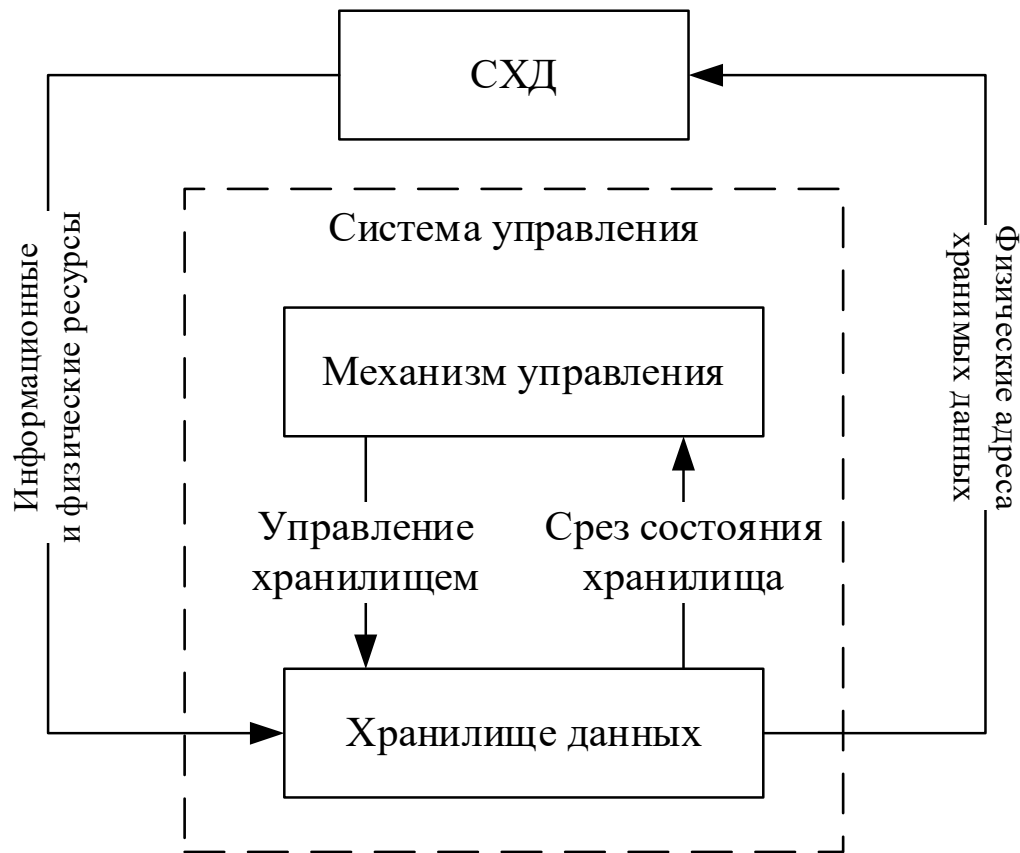


Рисунок 3 – Система управления физическим хранилищем данных

Представим СХД как систему управления физическим хранилищем данных

**Цель:** эффективное использование емкости хранилища, а также ее своевременное наращивание.

**Объект управления:** хранилище данных.

**Механизм управления:** совокупность алгоритмов управления хранилищем данных и составления прогноза его наращивания.

**Задачи системы управления:**

- распределение файлов в хранилище с максимальной эффективностью использования имеющейся емкости;
- прогнозирование расхода емкости для ее своевременного наращивания.

**Критерий эффективности:**

коэффициент  $k = \frac{V_{files}}{V_{data}}$  при  $k \rightarrow 1$

где  $V_{files}$  – это емкость, занимаемых файлами (с учетом пустых секторов диска);

$V_{data}$  – емкость, физически занимаемая данными.

# МЕХАНИЗМЫ УПРАВЛЕНИЯ

- механизм вертикального размещения – выбор уровня хранилища в зависимости от времени хранения;
- механизм горизонтального размещения – выбор файловой системы с определенным размером логического блока данных (для уровня RAID) или типа архивного носителя;
- механизм динамического размещения – механизм миграции данных по уровням в зависимости от частоты обращения к ним;
- прогнозирование расхода емкости хранилища для его своевременного наращивания.

# МЕХАНИЗМ ВЕРТИКАЛЬНОГО РАЗМЕЩЕНИЯ

Таблица 1

| ID                                | Типы данных (type)                       | Уровень хранилища             |
|-----------------------------------|--|-------------------------------|
| <i>ind</i> (initial data)         | Исходные данные                          | RAID                          |
| <i>bck</i> (backups)              | Резервные копии, архивные данные         | Автоматизированные библиотеки |
| <i>ngd</i> (next generation data) | Данные для передачи следующим поколениям | Носители длительного хранения |

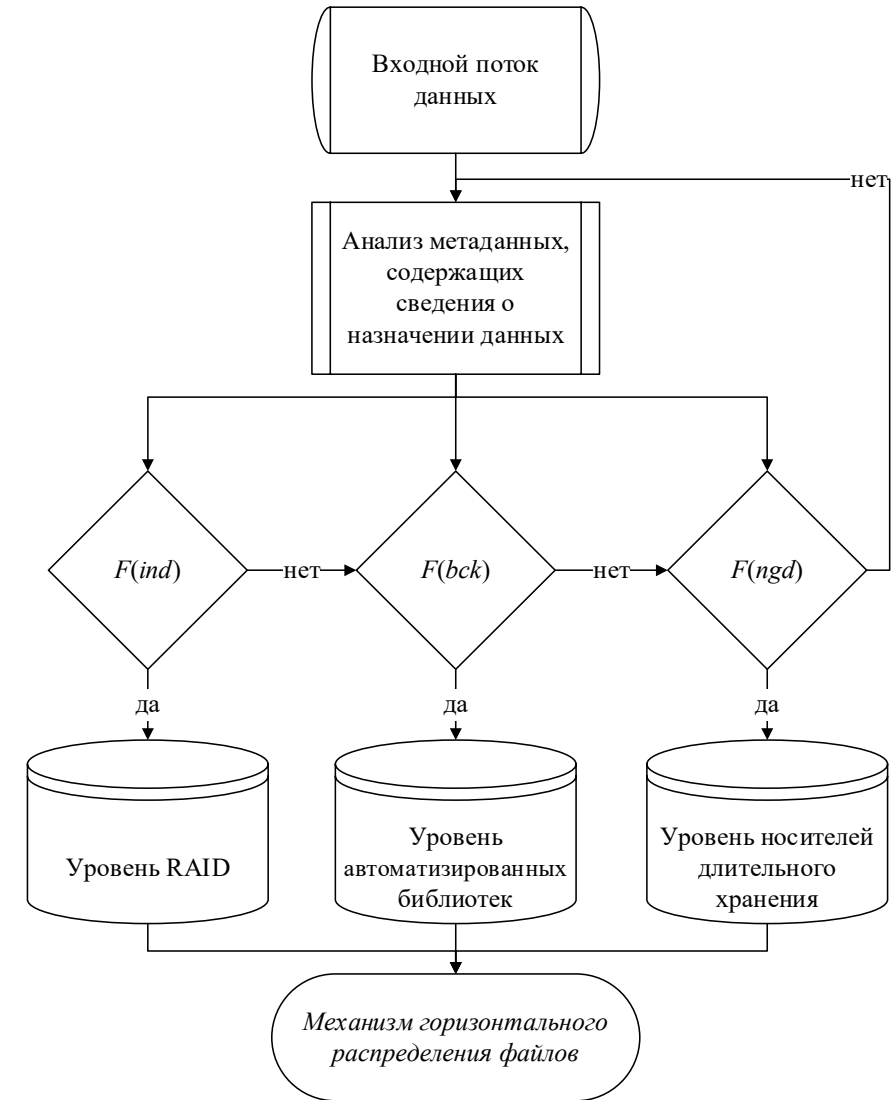


Рисунок 4 – Блок-схема алгоритма вертикального размещения

# МЕХАНИЗМ ГОРИЗОНТАЛЬНОГО РАЗМЕЩЕНИЯ

Таблица 2

| Пороговое значение размера файла ( $f$ ) | Размер логического блока данных | Том дискового хранилища |
|--|---------------------------------|-------------------------|
| $f_1$                                    | $a_1$                           | Том <sub>1</sub>        |
| $f_2$                                    | $a_2$                           | Том <sub>2</sub>        |
| ...                                      | ...                             | ...                     |
| $f_n$                                    | $a_n$                           | Том <sub>n</sub>        |

При этом:  $f_1 < f_n; a_1 < a_n$ .

Таблица 3

| Пороговое значение размера файла ( $f$ ) | Пример носителя |
|--|-----------------|
| Уровень автоматизированных библиотек     |                 |
| $f_1$                                    | $al_1$          |
| $f_2$                                    | $al_2$          |
| ...                                      | ...             |
| $f_n$                                    | $al_n$          |
| Уровень носителей длительного хранения   |                 |
| $f_1$                                    | $lt_1$          |
| $f_2$                                    | $lt_2$          |
| ...                                      | ...             |
| $f_n$                                    | $lt_n$          |

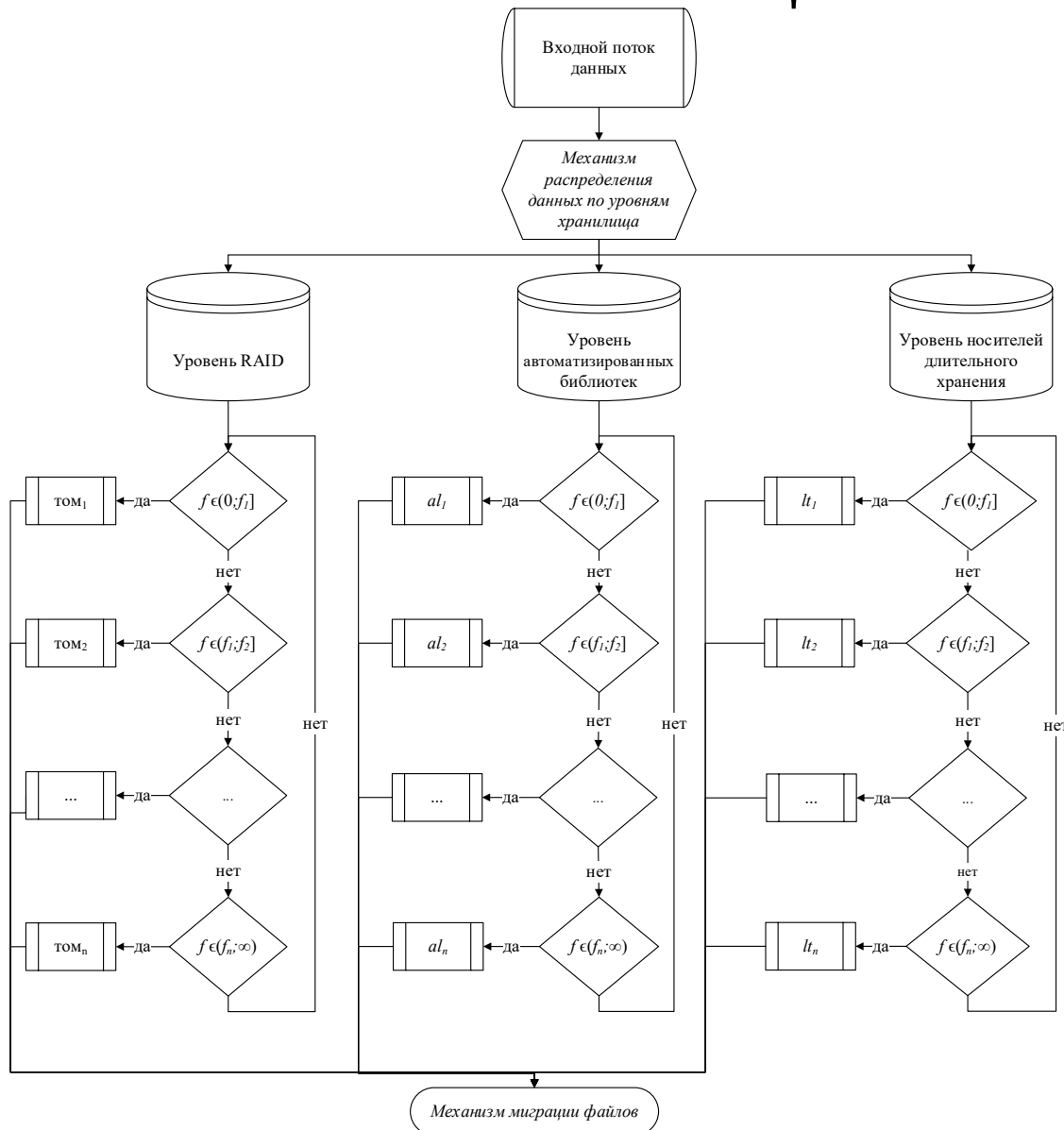


Рисунок 5 – Блок-схема алгоритма горизонтального размещения

# МЕХАНИЗМ ДИНАМИЧЕСКОГО РАЗМЕЩЕНИЯ

Таблица 4

| Частота обращения к файлам           | Уровень хранилища             |
|--------------------------------------|-------------------------------|
| $\lambda \in (\lambda_1; \infty)$    | RAID                          |
| $\lambda \in (\lambda_2; \lambda_1]$ | Автоматизированные библиотеки |
| $\lambda \in [0; \lambda_2]$         | Носители длительного хранения |

где  $\lambda_1$  и  $\lambda_2$  – пороговые значения частоты.

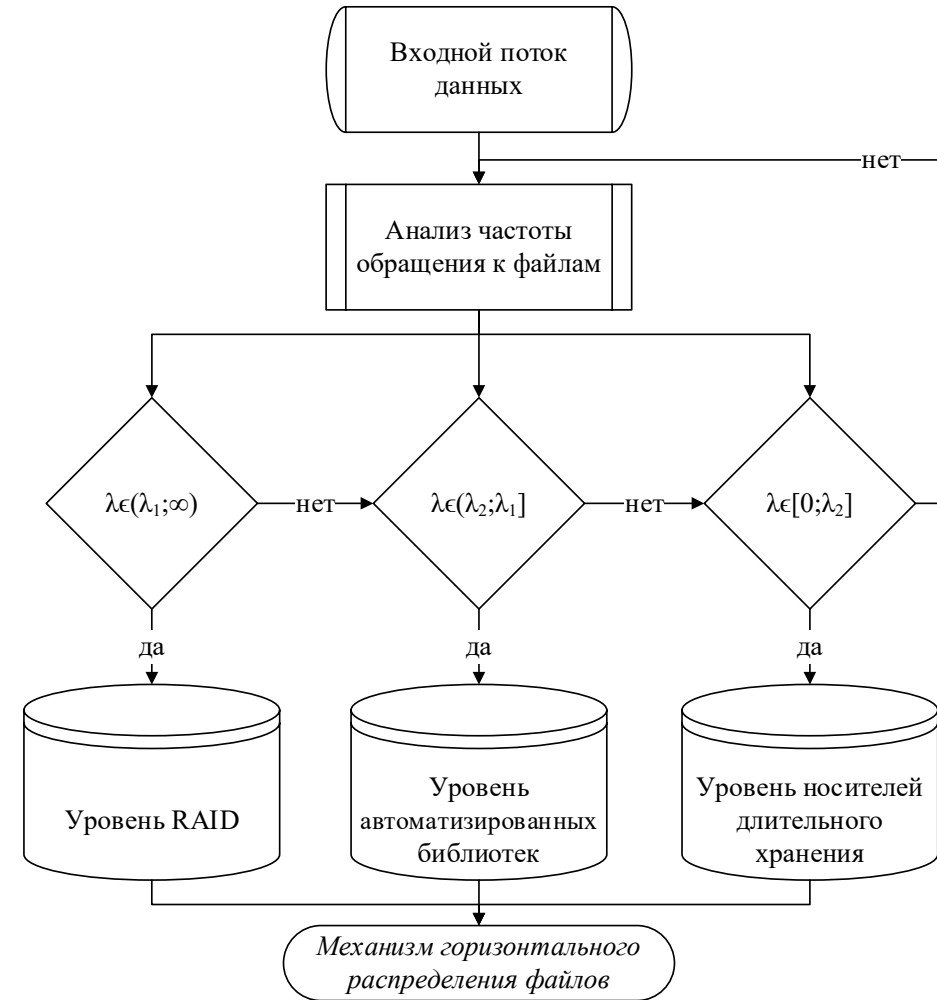


Рисунок 6 – Блок-схема алгоритма динамического размещения



# МЕХАНИЗМ НАРАЩИВАНИЯ ЕМКОСТИ

|   |   |   |
|---|---|---|
| $type_1,$<br>$V_{max11}, V_{lim11},$<br>$V_{current11}$<br>$t_{max11}, t_{lim11}$ | $type_1,$<br>$V_{max12}, V_{lim12},$<br>$V_{current12}$<br>$t_{max12}, t_{lim12}$ | $type_1,$<br>$V_{max13}, V_{lim13},$<br>$V_{current13}$<br>$t_{max13}, t_{lim13}$ |
| $type_2,$<br>$V_{max21}, V_{lim21},$<br>$V_{current21}$<br>$t_{max21}, t_{lim21}$ | $type_2,$<br>$V_{max22}, V_{lim22},$<br>$V_{current22}$<br>$t_{max22}, t_{lim22}$ | $type_2,$<br>$V_{max23}, V_{lim23},$<br>$V_{current23}$<br>$t_{max23}, t_{lim23}$ |
| $type_3,$<br>$V_{max31}, V_{lim31},$<br>$V_{current31}$<br>$t_{max31}, t_{lim31}$ | $type_3,$<br>$V_{max32}, V_{lim32},$<br>$V_{current32}$<br>$t_{max32}, t_{lim32}$ | $type_3,$<br>$V_{max33}, V_{lim33},$<br>$V_{current33}$<br>$t_{max33}, t_{lim33}$ |

Рисунок 7 – Паттерн поведения

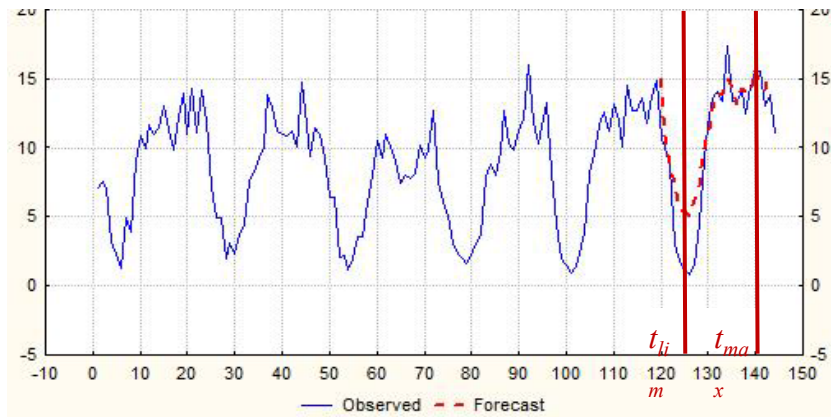


Рисунок 8 – Время преодоления пограничных значений емкости

- $V_{max}$ , максимальное значение емкости;
- $V_{lim}$  ( $V_{lim} < V_{max}$ ) - предельное значение емкости, при достижении которого необходимо производить наращивание;
- $\lambda_1, \dots, \lambda_{m-1}$  - пограничные значения частоты обращения к файлам при преодолении которых осуществляется миграция файлов по уровням системы.
- $V_{maxmn}$  - максимальная емкость ячейки  $mn$ ;
- $V_{limmn}$  - предельная емкость ячейки  $mn$ ;
- $V_{currentmn}$  текущая емкость ячейки  $mn$ ;
- $t_{limmn}$  - время достижения предельной емкости ячейки  $mn$ ;
- $t_{maxmn}$  - время достижения максимальной емкости ячейки  $mn$ .

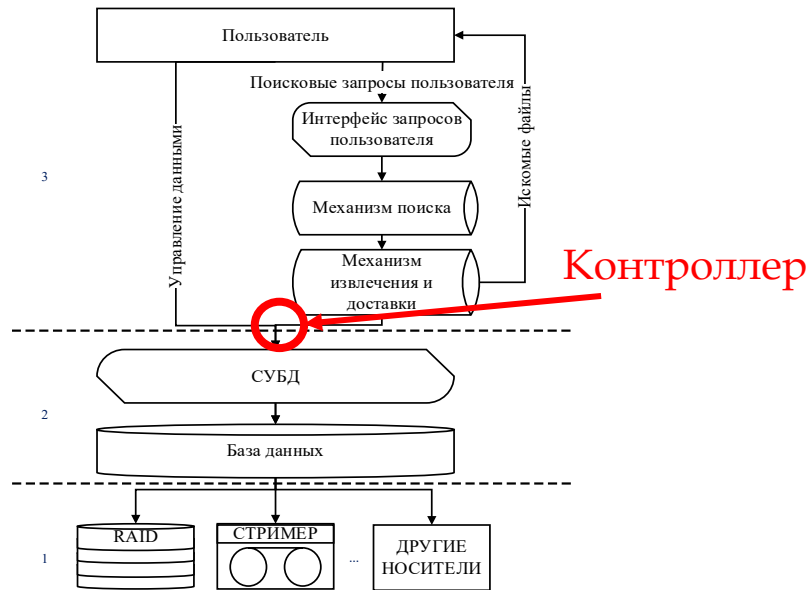
$$V_{lim} = \int_1^{t_{lim}} f(t)dt = h \sum_1^{t_{lim}} f(t), \quad (1)$$

$$V_{max} = \int_1^{t_{max}} f(t)dt = h \sum_1^{t_{max}} f(t), \quad (2)$$

где  $f(t)$  - функция входящего потока данных.  
 $h$  - шаг разбиения, равный единице минимального выбранного масштаба времени.

**Задача прогнозирования** заключается в нахождении времени достижения предельной емкости  $t_{limmn}$  и времени достижения максимальной емкости  $t_{maxmn}$  каждой ячейки.

# Вывод



Для реализации предложенных **механизмов размещения и миграции** необходим некий программируемый логический контроллер, работающий по предложенным алгоритмам и осуществляющий автоматическое распределение и миграцию данных внутри физического хранилища.

Данный контроллер связан непосредственно с физическим хранилищем данных и не зависит от логической структуры базы данных и используемой СУБД.

Рисунок 9 – Иерархическая структура СХД с контроллером

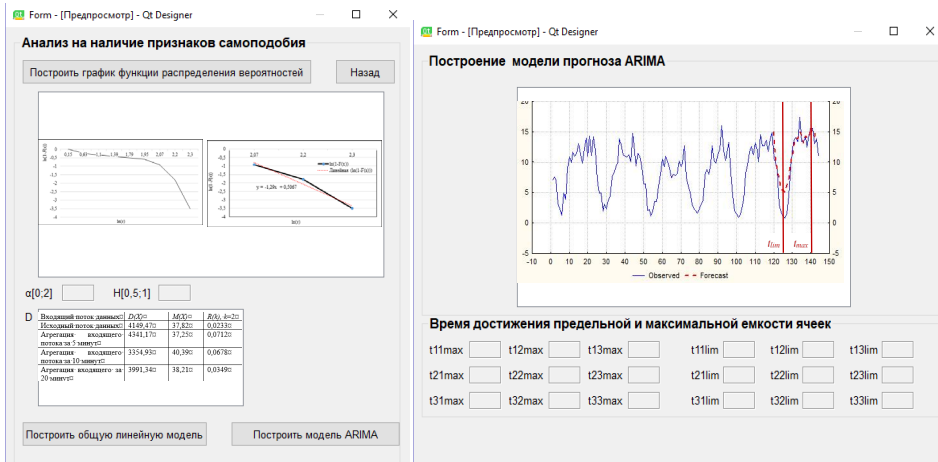


Рисунок 10 – Примеры окон интерфейса

Для реализации **механизма прогноза** было разработано приложение, помогающее автоматизировать процесс прогнозирования расхода емкости ячеек матрицы хранения.