Цель разработки оптимизированный алгоритм и аналитическая модель для системы создания планов телескопов — увеличение количества объектов, которые можно пронаблюдать за ночь.

Данная работа посвящена разработке аналитической модели, которая позволит оптимизировать выполнение задачи по наблюдению объектов с помощью оптических средств средней и малой апертуры.

Разработанная модель должна предоставить возможность оптимизации последовательности действий для телескопа загружая время работы до максимальных значений.

Данная АС является актуальной, потому что позволяет спланировать большое количество аппаратов, доступных для наблюдения, организовать собственный список.

Использование классов CIF в решении данной задачи, позволит снизить нагрузку на вычислительный агрегат, увеличить пропускную способность, увеличить точность и процент заполнения рабочего времени.

Разработанная система позволяет заполнить время работы оптического средства до максимально допустимых значений.

Оптимизация алгоритма

Библиотека для реализации

Анализ необходимых параметров системы

Анализ формирования программных файлов

Доработка системы для использования с Web-сайта

Задачи наблюдения за космическими объектами появились совсем недавно, около 10 лет назад, но уже существует 4 общеизвестных аналога, лидером обнаружения среди российских космических агенств является АСПОС, среди американских – NASA. [10][11]

Таблица 1. Сравнение аналогов и прототипов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерий | «АС ПРТ АОП» | SSPS | NASA | АСПОС |
| Простота | простая | простая | средняя | сложная |
| Уровень автоматизации процессов | высокий | высокий | высокий | средний |
| Добавление новых средств | + | + | + | + |
| Фильтрация объектов | + | + | + | + |
| Скорость работы | высокая | высокая | высокая | средняя |
| Графический анализ работы телескопа | + | – | – | + |
| Динамическое добавление объектов | + | – | – | + |

Таблица 2. Шкала присуждения баллов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оценка | Отлично | Хорошо | Удовлетворительно | Отсутствует |
| Балл | 1 | 0,7 | 0,3 | 0 |

Таблица 3. Сравнение аналогов и прототипов с учетом баллов и весовых коэффициентов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | Коэффициент | «АС ПРТ АОП» | SSPS | NASA | АСПОС |
| Простота | 0,075 | 1 | 1 | 0,7 | 0,3 |
| Уровень автоматизации процессов | 0,1 | 0,7 | 1 | 1 | 0,7 |
| Поиск по каталогу | 0,15 | 0,7 | 0,7 | 0,7 | 0,3 |
| Добавление новых средств | 0,15 | 0,7 | 0,3 | 0,3 | 0,7 |
| Скорость работы | 0,075 | 0,7 | 1 | 1 | 0,3 |
| Графический анализ работы телескопа | 0,25 | 0,7 | 0 | 0 | 0,3 |
| Динамическое добавление объектов | 0,2 | 0,7 | 0 | 0 | 0,7 |
| **Итого** | 1 | 0,72 | 0,40 | 0,38 | 0,48 |

АС ПРТ АОП имеет наибольшее количество баллов, так как набрала лучший балл из представленных аналогов по наиболее важным критериям: скорость работы, загрузка рабочего времени, отказоустойчивость системы, удобство работы, документация, возможность расширения функционала. Таким образом, АС ПРТ АОП является лучшей системой среди аналогов и прототипов, тем самым оправдывая свое создание.[12][13][14]

* + 1. **Выбор web-сервера**

Для моей системы будет использоваться:

* web-сервер, чтобы принимать запросы от пользователя;
* сервер-приложений, поскольку web-сервер самостоятельно не может самостоятельно формировать план;
* статистические файлы будут храниться в распределенном файловом хранилище, что намного масштабируемо и универсально, чем просто хранить на диске web-сервера.[15][16]

В качестве веб-сервера был выбран **apache**, поскольку также является Open Source продуктом. Кроме того, **apache** использует асинхронную систему ввода-вывода информации, что на практике означает экономию ресурсов системы и выгодно сказывается в случае больших нагрузок, обладает встроенной устойчивостью к многим видам DoS и DDos атак.[17][18]

**Описание apache**

Apache — это HTTP сервер, обладающий высокой надежностью и гибкостью, под HTTP сервером следует понимать программное обеспечение для обработки HTTP запросов. Основная работа Apache это обработка и ответ на HTTP запросы и генерирование динамическое содержание страниц. Гибкость достигается путем использования файла .htaccess, благодаря которому можно переопределять глобальные настройки сервера Apache. Выглядит это следующим образом.[19][20] Apache имеет свои глобальные настройки, но при обращении к сайту он ищет файл .htaccess, считывает из него директивы и применяет их, а не глобальные. Обычно файл .htaccess лежит в корне сайта, и определяет настройки сервера для всего сайта, однако он может располагаться и в любом внутреннем каталоге сайта, определяя таким образом настройки сервера именно для этого каталога. Вебмастерами этот файл используется для настройки переадресации, обработки ошибок, безопасности, доступа, кодировки и т.д. Apache имеет механизм для определения виртуальных хостов, благодаря чему на одном IP адресе может находится неограниченное количество сайтов, именно таким образом работает виртуальный хостинг. Также существует большое количество модулей, позволяющих ему работать с большинством популярных языков программирования и расширять функциональность.

### **Выбор средства разработки**

В качестве средства разработки был выбран Python. Python – интерпретируемый язык программирования. Python — это свободный интерпретируемый объектно-ориентированный расширяемый встраиваемый язык программирования очень высокого уровня. – свободный — все исходные тексты интерпретатора и библиотек доступны для любого, включая коммерческое, использования; – интерпретируемый — потому что использует “позднее связывание”; – объектно-ориентированный — классическая ОО модель, включая множественное наследование; – расширяемый — имеет строго определенные API для создания модулей, типов и классов на C или C++; – встраиваемый — имеет строго определенные API для встраивания интерпретатора в другие программы; – очень высокого уровня — динамическая типизация, встроенные типы данных высокого уровня, классы, модули, механизм исключений. Python — язык универсальный, он широко используется во всем мире для самых разных целей — базы данных и обработка текстов, встраивание интерпретатора в игры, программирование GUI и быстрое создание прототипов (RAD). И, конечно же, Python используется для программирования Internet и Web приложений — серверных (CGI), клиентских (роботы), Web-серверов и серверов приложений. Python обладает богатой стандартной библиотекой, и еще более богатым набором модулей, написанных третьими лицами.

Python позволяет разбивать программы на модули, которые затем могут быть использованы в других программах. Python поставляется с большой библиотекой стандартных модулей, которые Вы можете использовать как основу для Ваших программ или в качестве примеров при изучении языка. Стандартные модули предоставляют средства для работы с файлами, системных вызовов, сетевые соединения и даже интерфейсы к различным графическим библиотекам. Python — интерпретируемый язык, что позволит Вам сэкономить значительное количество времени, обычно расходуемого на компиляцию. Интерпретатор можно использовать интерактивно, что позволяет экспериментировать с возможностями языка, писать наброски программ или тестировать функции при разработке “снизу вверх”. Он также удобен в качестве настольного калькулятора.

Python расширяемый: знание C позволит Вам добавлять новые встраиваемые функции или модули для выполнения критичных операций с максимальной скоростью или написания интерфейса к коммерческим библиотекам, доступным только в двоичном виде. Вы можете вставить интерпретатор языка Python в приложение, написанное на C, и использовать его в качестве расширения или командного языка для этого приложения.

При выборе языка программирования непосредственно для реализации алгоритма создания плана выбор пал на 3 самых быстрых и самых распространенных языка: Java, Python и С++. Между языками был проведен анализ в скорости выполнения. [3] в итоге был выбран язык С++.

Язык C++ представляет собой набор команд, которые говорят компьютеру, что необходимо сделать. Этот набор команд, обычно называется исходный код или просто код. Командами являются или «функции» или «ключевые слова». Ключевые слова(зарезервированные слова С/С++) являются основными строительными блоками языка. Функции являются сложными строительными блоками, так как записаны они в терминах более простых функций — вы это увидите в нашей самой первой программе, которая показана ниже. Такая структура функций напоминает содержание книги. Содержание может показывать главы книги, каждая глава в книге может иметь своё собственное содержание, состоящее из пунктов, каждый пункт может иметь свои подпункты. Хотя C++ предоставляет много общих функций и зарезервированных слов, которые вы можете использовать, все-таки возникает потребность в написании своих собственных функций.

Широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java и C#.

Синтаксис C++ унаследован от языка [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)). Одним из принципов разработки было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как [компиляторами](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C.

Стандартная библиотека C++ включает в себя набор средств, которые должны быть доступны для любой реализации языка, чтобы обеспечить программистам удобное пользование языковыми средствами и создать базу для разработки как прикладных приложений самого широкого спектра, так и специализированных библиотек. Стандартная библиотека C++ включает в себя часть стандартной библиотеки C.

Доступ к возможностям стандартной библиотеки C++ обеспечивается с помощью включения в программу (посредством директивы #include) соответствующих стандартных заголовочных файлов. Всего в стандарте C++11 определено 79 таких файлов. Средства стандартной библиотеки объявляются как входящие в пространство имён std. Заголовочные файлы, имена которых соответствуют шаблону «cX», где X — имя заголовочного файла стандартной библиотеки C без расширения (cstdlib, cstring, cstdio и пр.), содержат объявления, соответствующие данной части стандартной библиотеки C. Стандартные функции библиотеки C также находятся в пространстве имён std.

### **Выбор сервера приложений**

Наиболее полным и распространенным фреймворком для Python является **Django**. Именно ее будем использовать для разработки серверной стороны планировщика.

**Описание Django**

Django – [свободный](http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [фреймворк](http://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) для [веб-приложений](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) на языке [Python](http://ru.wikipedia.org/wiki/Python), использующий шаблон проектирования [MVC](http://ru.wikipedia.org/wiki/Model-View-Controller). Проект поддерживается организацией [Django Software Foundation](http://ru.wikipedia.org/w/index.php?title=Django_Software_Foundation&action=edit&redlink=1).

Архитектура Django похожа на «[Модель-Представление-Контроллер](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%9F%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5-%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80)» (MVC). Контроллер классической модели MVC примерно соответствует уровню, который в Django называется Представление ( View), а презентационная логика Представления реализуется в Django уровнем Шаблонов (Template). Из-за этого уровневую архитектуру Django часто называют «Модель-Шаблон-Представление» (MTV).

Django проектировался для работы под управлением [Apache](http://ru.wikipedia.org/wiki/Apache) с модулем [mod python](http://ru.wikipedia.org/wiki/Mod_python). С включением поддержки [WSGI](http://ru.wikipedia.org/wiki/WSGI), Django может работать под управлением [FastCGI](http://ru.wikipedia.org/wiki/FastCGI), [mod\_wsgi](http://ru.wikipedia.org/wiki/Mod_wsgi) или [SCGI](http://ru.wikipedia.org/wiki/SCGI) на [nginx](http://ru.wikipedia.org/wiki/Nginx) и других серверах ([lighttpd](http://ru.wikipedia.org/wiki/Lighttpd), [nginx](http://ru.wikipedia.org/wiki/Nginx) и пр.).

В настоящее время Django может работать с другими [СУБД](http://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94): [MySQL](http://ru.wikipedia.org/wiki/MySQL), [SQLite](http://ru.wikipedia.org/wiki/SQLite), [Microsoft SQL Server](http://ru.wikipedia.org/wiki/Microsoft_SQL_Server), [DB2](http://ru.wikipedia.org/wiki/DB2), [Firebird](http://ru.wikipedia.org/wiki/Firebird), SQL Anywhere, [PostgreSQL](http://ru.wikipedia.org/wiki/PostgreSQL) и [Oracle](http://ru.wikipedia.org/wiki/Oracle_%28%D0%A1%D0%A3%D0%91%D0%94%29).

В составе Django присутствует собственный [веб-сервер](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) для разработки. Сервер автоматически определяет изменения в файлах исходного кода проекта и перезапускается, что ускоряет процесс разработки на [Python](http://ru.wikipedia.org/wiki/Python), но пригоден только для процесса разработки и отладки приложения.

Некоторые возможности Django: Встроенный интерфейс администратора, с уже имеющимися переводами на многие языки;

* Диспетчер URL на основе [регулярных выражений](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F);
* Расширяемая система шаблонов с тегами и наследованием;
* Система кеширования;
* Библиотека для работы с формами (наследование, построение форм по существующей модели БД);

Встроенная автоматическая документация по тегам шаблонов и моделям данных, доступная через административное приложение.

### **Выбор распределенного файлового хранилища**

В качестве распределенного файлового хранилища был выбран Elliptics. Elliptics – это распределенное key -value-хранилище данных с открытым исходным кодом. При стандартном поведении представляет собой классическую DHT (распределенную хэш-таблицу). Не требует специальных управляющих узлов, поэтому не содержит единых точек отказа.

Elliptics позволяет распределять данные по неограниченному количеству реплик и между несколькими дата-центрами. В данном случае «реплика» — это группа, которая представляет собой DHT-кольцо с диапазоном ключей от 0 до 2^512.

В качестве транспорта используется p2p-протокол. Помимо этого Elliptics поддерживает полноценный сервер-сайд скриптинг (на Python, JavaScript и C++).

Для реализации данного продукта я создал граф диалога, на котором наглядно видно, какие действия может выполнять пользователь находясь на странице моего проекта. Имеется 4 основные варианта действий и 6 второстепенных от одного основного. Благодаря этой схеме значительно проще разобраться в работе системы.

На рисунке 1 изображен граф диалога, разработанного пользовательского интерфейса

### **Выбор программных средств**

В качестве сетевой операционной системы была выбрана Debian8,6. В качестве ОС для сервера как правило используются Windows, Novell NetWare, Unix, Linux. Однако выбор в качестве операционной системы Debianбыл обусловлен тем, что данная ОС, как и любая другая операционная система на базе Linux, также является Open Source продуктом и распространяется бесплатно. Debianимеет преимущества в производительности, менее требовательна к ресурсам.

Debian отличается широким спектром возможностей. В текущую стабильную версию включено свыше пятидесяти одной тысячи пакетов программ для десяти архитектур на основе ядра Linux (от Intel/AMD 32-bit/64-bit, широко применяемых в персональных компьютерах, до ARM, обычно используемых во встраиваемых системах и мейнфреймах IBM System) и также двух архитектур на основе ядра FreeBSD(kfreebsd-i386 and kfreebsd-amd64).

Отличительными чертами Debian являются: система управления пакетами Advanced Packaging Tool (APT), жёсткая политика по отношению к пакетам, репозитории с огромным их количеством, а также высокое качество выпускаемых версий. Это сделало возможным простое обновление между версиями, а также автоматическую установку и удаление пакетов. Именно в Debian впервые был введён как единый стандарт механизм выбора предпочтительного ПО среди нескольких вариантов — Alternatives.

Также Debian является единственным дистрибутивом со смешанной системой обновлений — как традиционная с жизненным циклом, так и модель Rolling release. Переключение между моделями реализовано по кодовым именам в APT. Для Rolling Release прописывается в репозиторий имя ветки, например stable, а для традиционного жизненного цикла — имя дистрибутива, например stretch.

При стандартной установке Debian используется среда рабочего стола GNOME, куда включён набор популярных программ, таких как LibreOffice, Firefox, почтовая программа Evolution, программы для записи CD/DVD, проигрыватели музыки и видео, программы для просмотра и редактирования изображений и программы для просмотра документов в формате PDF. Также есть образы Live-CD, собранные с KDE, Xfce, LXDE, MATE и Cinnamon. Установочные диски — в количестве пяти (Lenny) или восьми (Squeeze) DVD или более тридцати CD — содержат все доступные и не обязательно необходимые для стандартной установки пакеты. Для метода установки по сети используется CD, который меньше обычного установочного CD/DVD. Он содержит только то, что необходимо для запуска установщика и скачивания пакетов, выбранных в процессе установки посредством APT. Образы CD/DVD можно свободно скачать через BitTorrent, jigdo или купить у одного из поставщиков компакт-дисков Debian. Также возможна установка с USB-накопителя. Существуют специальные наборы программ Debian Pure Blend, ориентированные на группы пользователей с различным практическим опытом и интересами: Debian Junior для детей от 1 до 99, Debian-Edu для школ, Debian Science с группировкой по наукам, Debian-Med для практического и научного использования в сфере здравоохранения и другие.

### **Выбор аппаратных средств**

Для оператора планировщика достаточны следующие минимальные характеристики:

* Процессор Pentium с частотой 1 ГГц или более быстрый
* Не менее 512 МБ оперативной памяти (рекомендуется не менее 1 ГБ);
* Видеоадаптер и монитор, способные обеспечить графический режим 800\*600 точек с 16 битной цветопередачей;
* Наличие манипулятора «мышь» или другого указывающего устройства;
* Наличие клавиатуры.

**Технические требования к серверу:**

* ПроцессорIntel Xeon E5-2640 v3 (8 Core) с тактовой частотой процессора 2.6 ГГц, 6 ядер, кэш 20 Мб;
* Чипсет DDR3, RAID 1 или 5, Ethernet 1000 Мбит/с;
* Оперативная память с объемом 64 Гб ECC, DDR3, 1600 МГц;
* Жёсткие диски объемом 1 Тб (RAID15), 7200 об/мин или SSD.

### **Структура системы и граф диалога алгоритма**

В рамках проекта по созданию системы создания планов был создан планировщик.

состоит из следующих модулей:

1. Модуль добавления средства.
2. Модуль удаления средства.
3. Модуль информации о средстве.
4. Модуль корректировки данных выбранного средства.
5. Модуль добавления собственного списка объектов.
6. Модуль корректировки данных получаемой информации.
7. Модуль приоритизации данных.
8. Модуль создания плана.
9. Модуль графического анализа.
10. Модуль используемых объектов.
11. Модуль незадействованных объектов.
12. Модуль создания плана для передачи на средство.

Структурная схема представлена на рисунке:

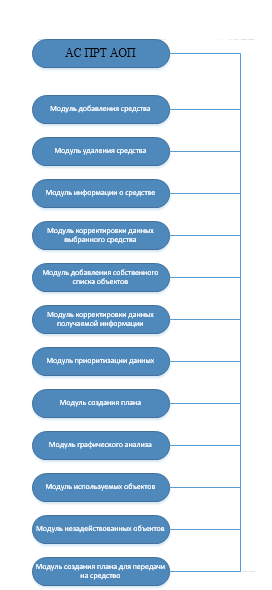


Рисунок 1.

На рисунке 2 представлен графи диалога системы планирования пользовательской визуализации. На этом графе

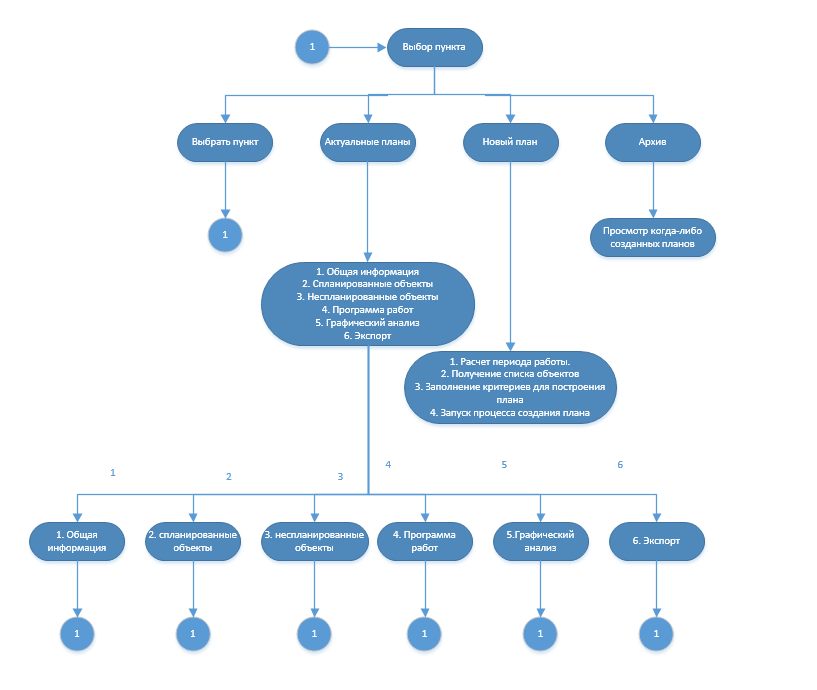


Рисунок 2. Граф диалога системы планирования