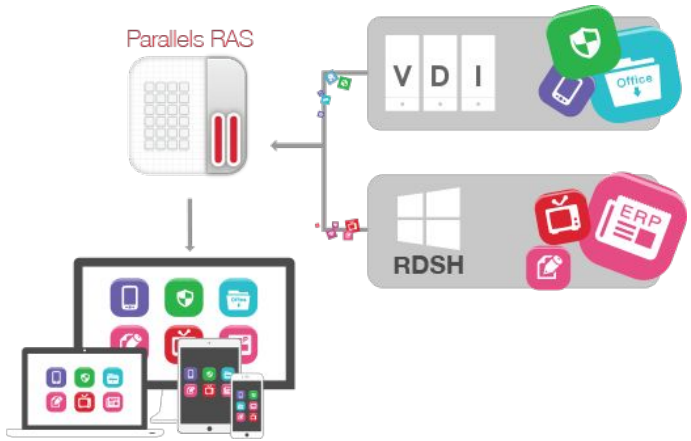
Балансировка ресурсов при публикации приложений

Выполнил: Ковалев В.В.
Научный руководитель: к.ф.-м.н. Кудрин М.Ю.

Существующие проекты

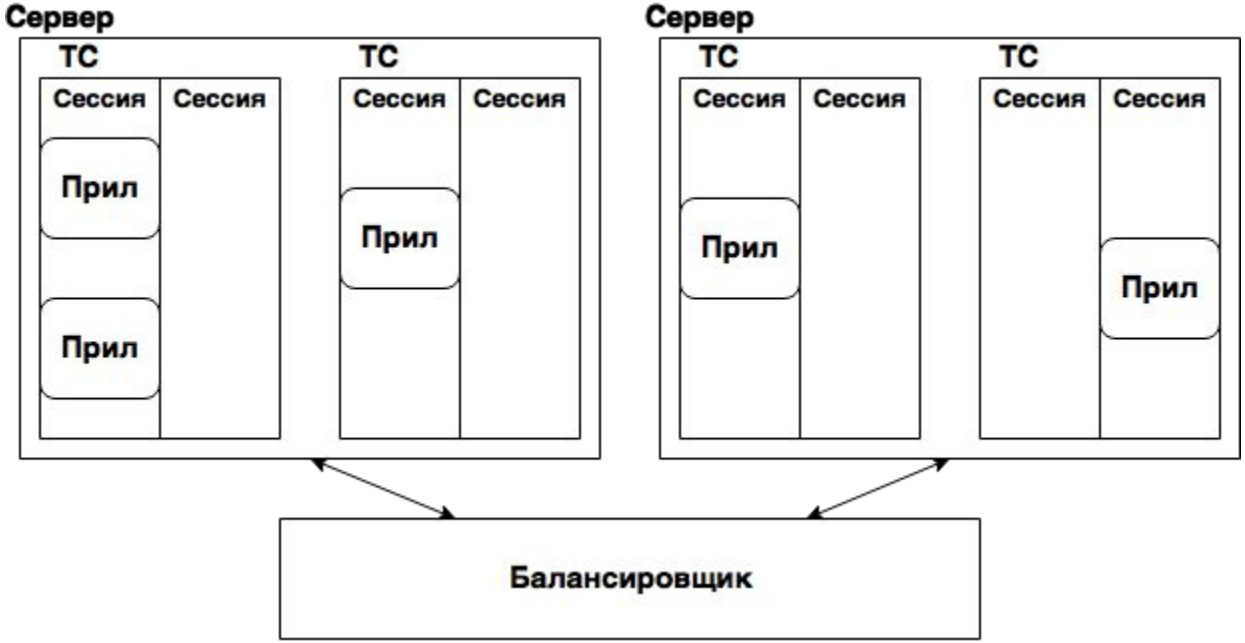
- XenDesktop (Citrix Systems)
- Horizon (VMware)
- Remote Desktop Services (Microsoft)
- Wyse vWorkspace (Dell)
- Remote Application Server (Parallels)



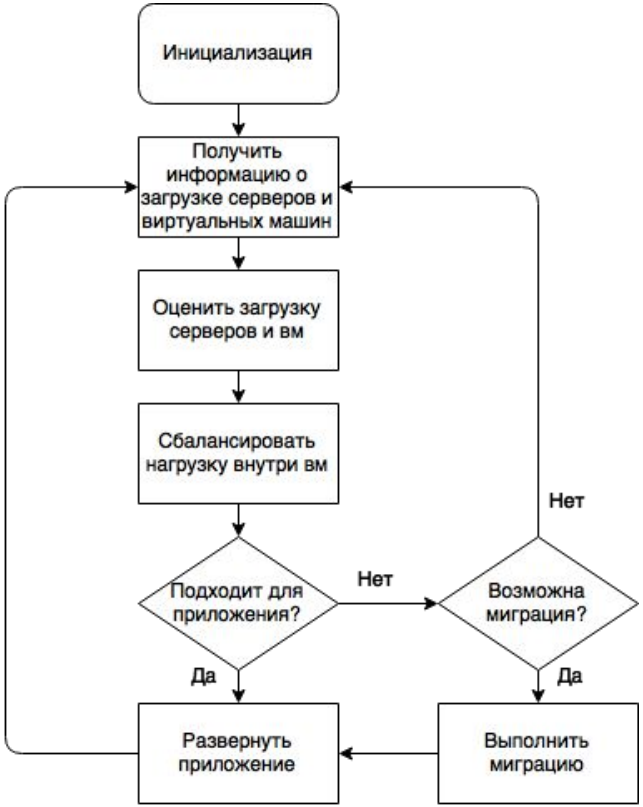
Преимущества

- Сокращение затрат на инфраструктуру
- Экономия времени
- Безопасность
- Гибкая масштабируемость

Схема работы системы при публикации приложений



Основная схема существующих продуктов



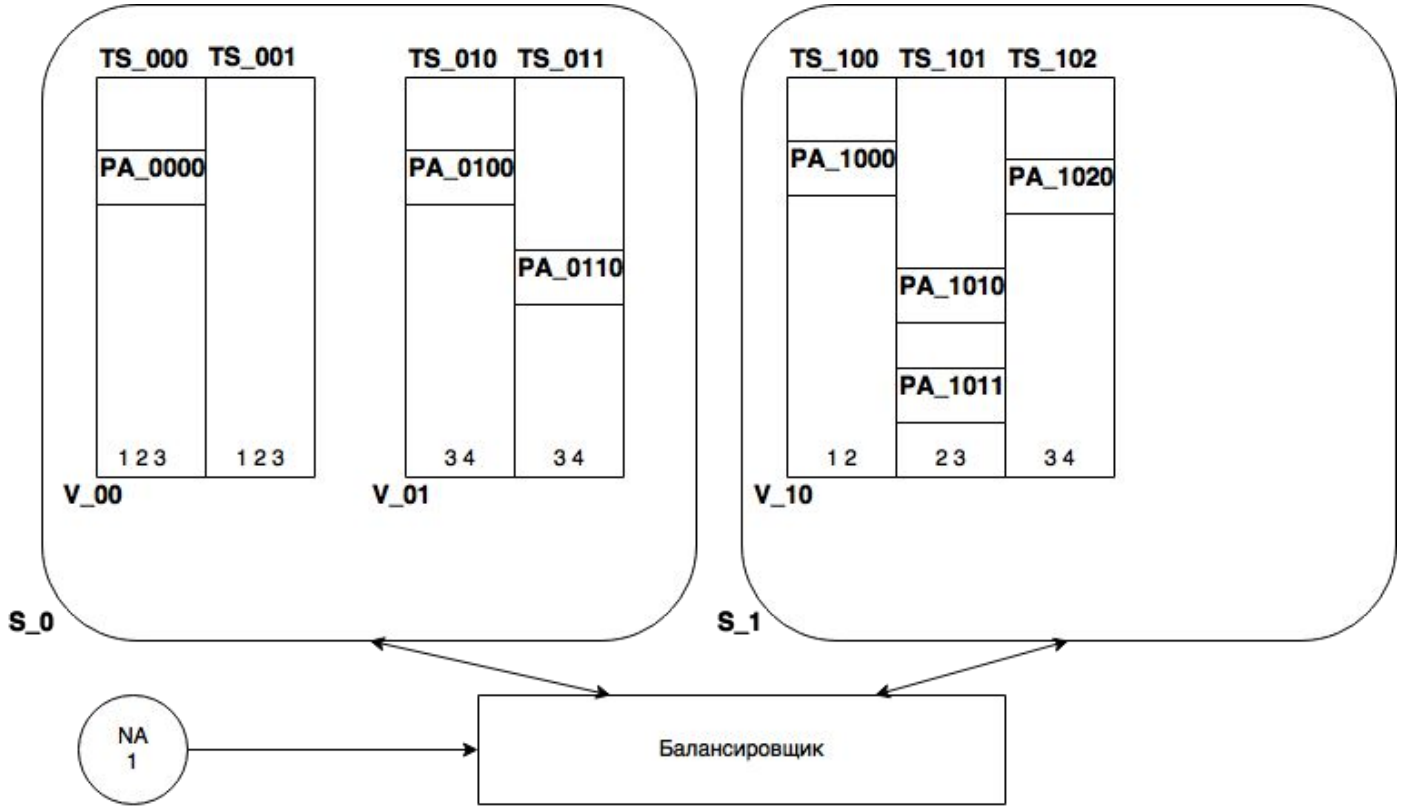
Поставленные задачи

1. Анализ существующих алгоритмов
2. Формулировка описания системы в случае публикации приложений
3. Математическая модель и формализация алгоритма
4. Реализация прототипа

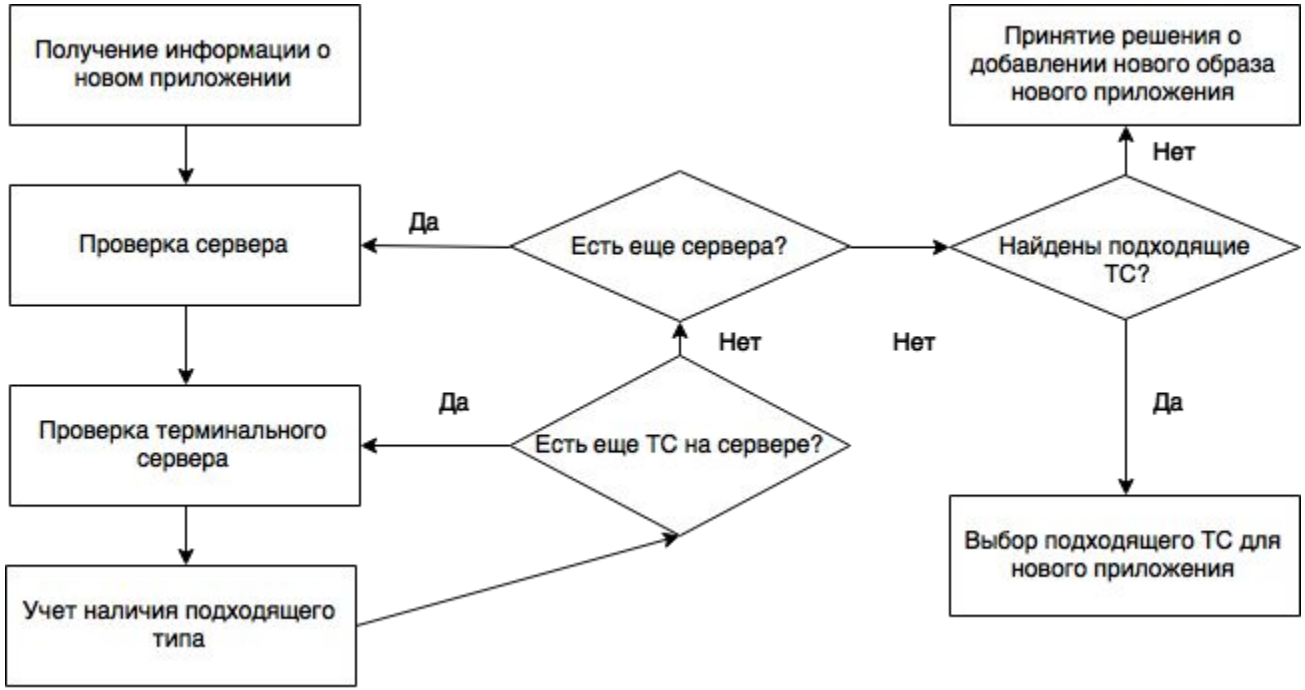
Поставленные задачи перед алгоритмом

- Где развернуть новое приложение?
- Учет типа приложения (не на каждой виртуальной машине это возможно)
- Выполнять ли миграцию виртуальной машины при решении задачи?

Схема работы



Учет типа нового приложения для публикации



Математическая модель

Object (CPU, Memory, Network, Threshold, Utilization, Weight)

Объектами являются S (Server), V (Terminal Server), TS (Terminal Session), PA (Published Application), NA (New Application)

NA имеет тип приложения, характеризующийся числом. Каждая V поддерживает определенный тип приложений, поэтому развернуть NA можно не везде. Уже опубликованное приложение (PA) имеет коэффициент, который характеризует текущую загруженность, U (Utilization). $0 <= U <= 1$.

S, V, TS имеют коэффициент T (Threshold), характеризующий максимальную практическую загруженность. Он введен, т.к. объекты сами по себе потребляют ресурсы + не рекомендуется загружать их полностью. $0 <= T <= 1$.

Математическая модель

$FL(res_i)$ - функция, рассчитывающая полезную нагрузку (free load) ресурса res для i-го сервера.

$$FL(res_i) = res_i * T_i \left(1 - \sum_{k=0}(W_{ik} T_{ik} \sum_{j=0}(W_{ikj} T_{ikj} \sum_{t=0}(W_{ikjt} U_{ikjt})))\right)$$

$$FL(res_{ikj}) = res_{ikj} * T_{ikj} \left(1 - \sum_{t=0}(W_{ikjt} U_{ikjt})\right)$$

Где i - индекс сервера. k - виртуальной машины. j - терминальной сессии.

t - опубликованного приложения.

Случай миграции

В ситуации, когда необходимо выполнить миграцию виртуальной машины, используется следующая формула:

$$\text{Min}\left\{\frac{FL(CPU_s)}{CPU_v}+\frac{FL(MEM_s)}{MEM_v}+\frac{FL(NET_s)}{NET_v}\right\},\left\{\begin{array}{l}FL(CPU_s) > CPU_v\\ FL(MEM_s) > MEM_v\\ FL(NET_s) > NET_v\end{array}\right.$$

Алгоритм определения места для публикации

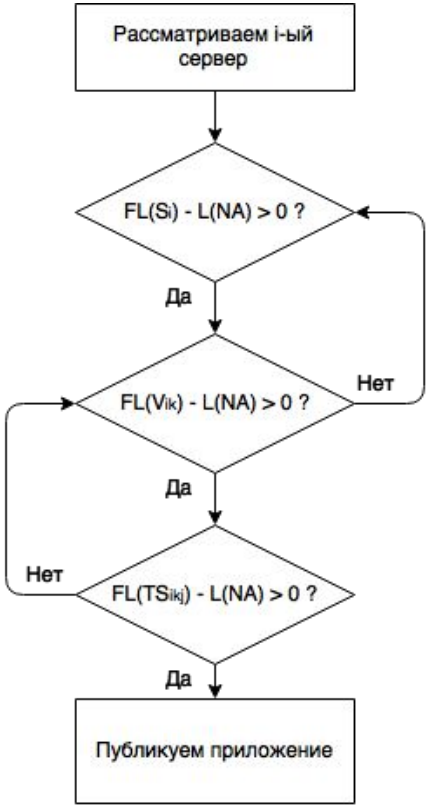
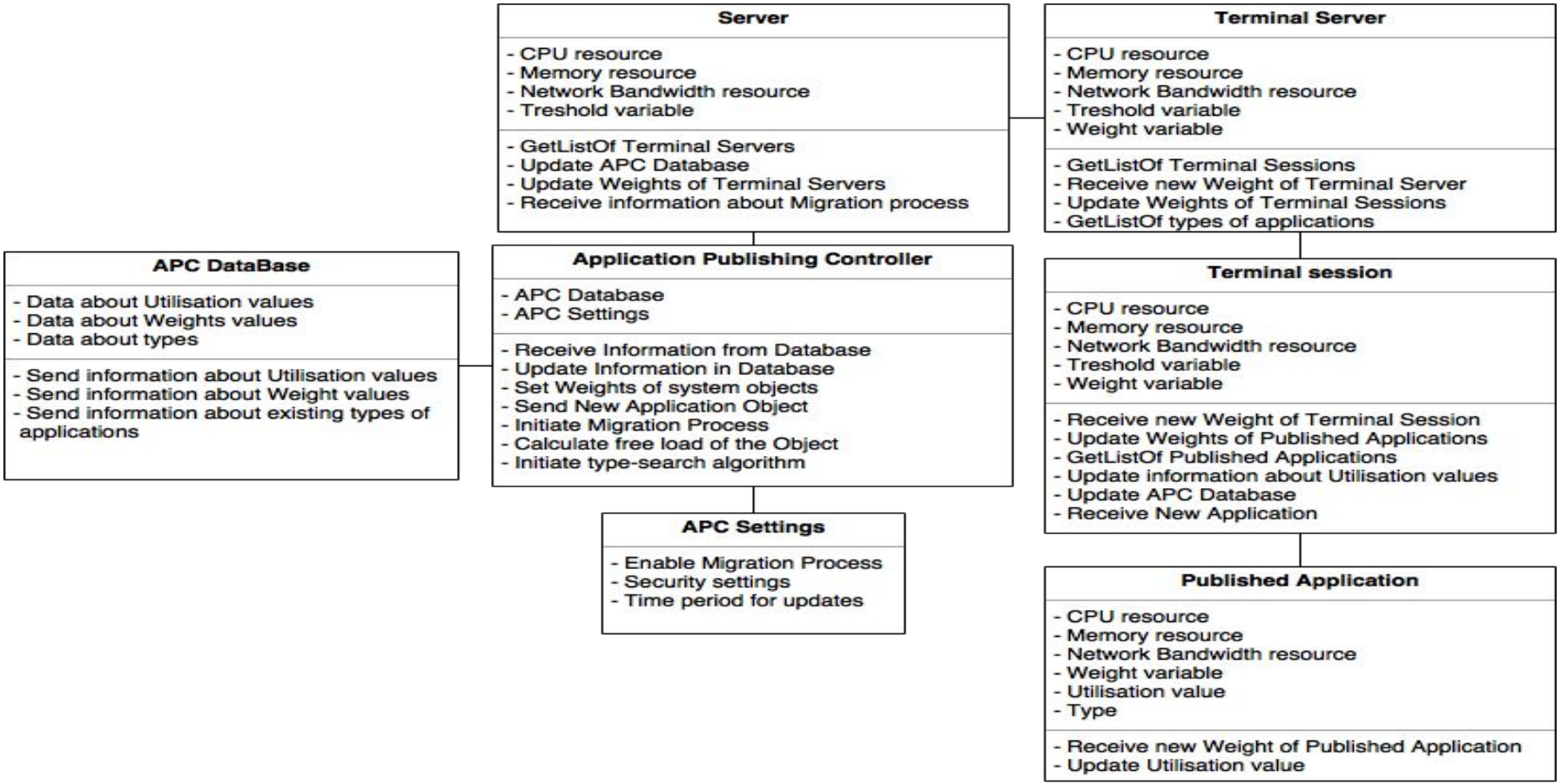


Схема реализации



Результаты проведенной работы

- Произведен анализ существующих решений
- Построена математическая модель
- Построен алгоритм и схема реализации
- Начато создание прототипа

Перспективы

- Учет рисков
- Учитывание временных задержек
- Дополнить различными сценариями поведения системы
- Внедрение реализации в Parallels Remote Application Server

Спасибо за внимание