## Балансировка ресурсов при публикации приложений

Выполнил: Ковалев В.В.

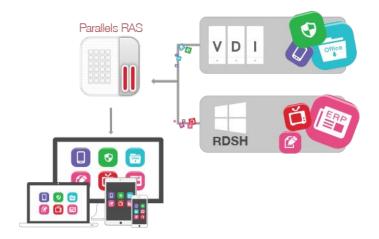
Научный руководитель: к.ф.-м.н. Кудрин М.Ю.

#### Существующие проекты

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



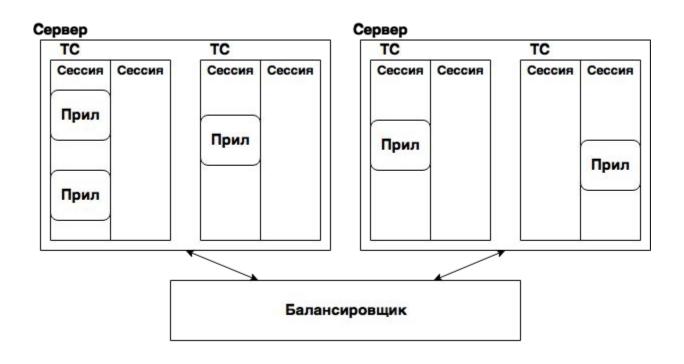




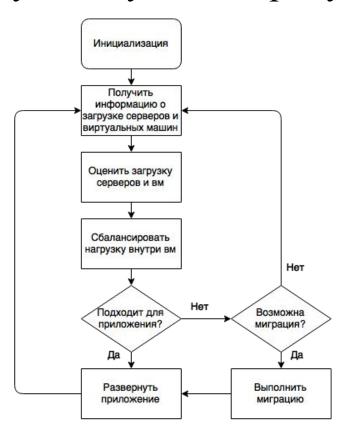
### Преимущества

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

#### Схема работы системы при публикации приложений



## Основная схема существующих продуктов



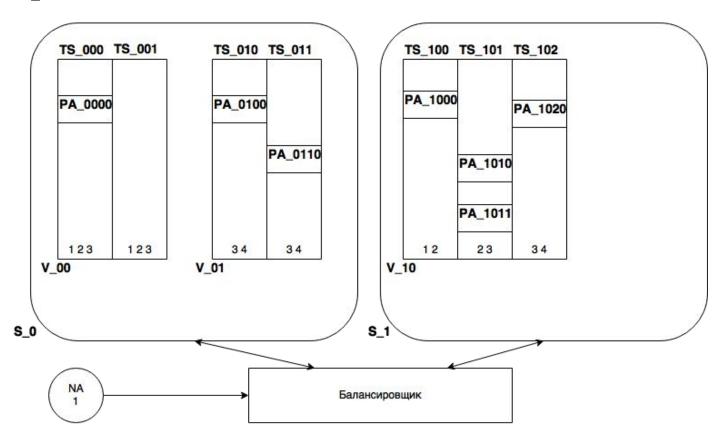
#### Поставленные задачи

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

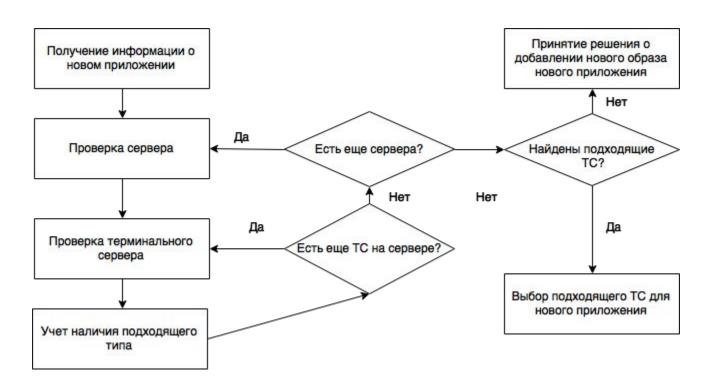
#### Поставленные задачи перед алгоритмом

- 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).  $0 \le U \le 1$ .

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.  $0 \le T \le 1$ .

#### Математическая модель

FL(res<sub>i</sub>) - функция, рассчитывающая полезную нагрузку (free load) ресурса res для i-го сервера.

$$FL(res_{_{i}}) = res_{_{i}} * T_{_{i}} (1 - \sum_{k=0} (W_{ik}T_{ik} \sum_{j=0} (W_{ikj}T_{ikj} \sum_{t=0} (W_{ikjt}U_{ikjt}))) \ )$$

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

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

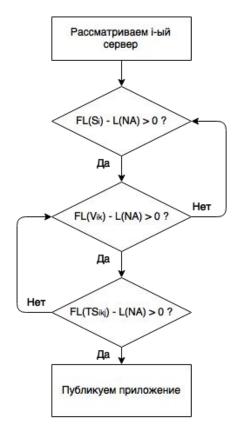
t - опубликованного приложения.

#### Случай миграции

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

$$\min \left\{ \frac{\text{FL(CPU_s)}}{\text{CPU_v}} + \frac{\text{FL(MEM_s)}}{\text{MEM_v}} + \frac{\text{FL(NET_s)}}{\text{NET_v}} \right\}, \begin{cases} \text{FL(CPU_s)} > \text{CPU_v} \\ \text{FL(MEM_s)} > \text{MEM_v} \\ \text{FL(NET_s)} > \text{NET_v} \end{cases}$$

#### Алгоритм определения места для публикации



#### Схема реализации

APC DataBase

Send information about Utilisation values

- Send information about existing types of

Send information about Weight values

Data about Utilisation values

Data about Weights values

Data about types

applications

#### Server CPU resource Memory resource Network Bandwidth resource Treshold variable GetListOf Terminal Servers Update APC Database Update Weights of Terminal Servers Receive information about Migration process Application Publishing Controller APC Database APC Settings Receive Information from Database Update Information in Database Set Weights of system objects - Send New Application Object - Initiate Migration Process - Calculate free load of the Object Initiate type-search algorithm **APC Settings** Enable Migration Process Security settings - Time period for updates

# Terminal Server - CPU resource - Memory resource - Network Bandwidth resource - Treshold variable - Weight variable - GetListOf Terminal Sessions - Receive new Weight of Terminal Server - Update Weights of Terminal Sessions - GetListOf types of applications

# Terminal session - CPU resource - Memory resource - Network Bandwidth resource - Treshold variable - Weight variable - Receive new Weight of Terminal Session - Update Weights of Published Applications - GetListOf Published Applications - Update information about Utilisation values - Update APC Database - Receive New Application

## Published Application - CPU resource - Memory resource - Network Bandwidth resource - Weight variable - Utilisation value - Type - Receive new Weight of Published Application - Update Utilisation value

#### 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