

Высокопроизводительный SA

Общие задачи

Описание проблемы

Общие предположения

Требования

#	Требования	User Story	Важность	Примечания
1				

Общий дизайн

Коллекция Scripts

Таблица sa_martask заменяется на коллекцию nos.scripts в mongodb. Каждый скрипт, запланированный к выполнению, оформляется как отдельный документ.

Поля документа

Поле	Тип	Описание
id	ObjectID	ID задачи
group	ObjectID	ID группы задач, если задан
timestamp	DateTimeField	Время создания скрипта
expires	DateTimeField	Крайний срок выполнения скрипта
status	StringField	Статус задачи. Значения: <ul style="list-style-type: none">W - waitingF - failedR - runningC - complete
result_code	IntField	Код завершения скрипта
max_retries	IntField	Максимальное количество попыток
retries	IntField	Текущее количество попыток
object	ForeignKeyField	Ссылка на ManagedObject
script	StringField	Название скрипта

[Translate to English](#)

СВОЙСТВА

Epic	
Type	FEATURE
Status	DRAFT
Open Date	2014-01-29
Modules	SA
Language	Russian
Owner	
Designer	Unknown User (dv)
Developers	Unknown User (dv)
QA	
Close Date	

Оглавление


- [Общие задачи](#)
- [Описание проблемы](#)
- [Общие предположения](#)
- [Требования](#)
- [Общий дизайн](#)
 - [API для запуска скрипта](#)
 - [API для запуска группы задач](#)
 - [SAService \(/SAService/\)](#)
- [Открытые вопросы](#)
- [Related Issues](#)
- [Отложенные задачи](#)
- [Свойства](#)
- [Оглавление](#)
- [История изменений](#)

История изменений

Version	Published	Сf	By
---------	-----------	----	----

access_profile	EmbeddedDocumentField	Параметры доступа. Вложенный документ вида																														
		<table border="1"> <thead> <tr> <th>Поле</th> <th>Тип</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>address</td> <td>StringField</td> <td>Адрес объекта</td> </tr> <tr> <td>port</td> <td>IntegerField</td> <td>Порт объекта</td> </tr> <tr> <td>profile</td> <td>StringField</td> <td>Профиль объекта</td> </tr> <tr> <td>user</td> <td>StringField</td> <td>CLI user</td> </tr> <tr> <td>password</td> <td>StringField</td> <td>CLI password</td> </tr> <tr> <td>super_password</td> <td>StringField</td> <td>enable password</td> </tr> <tr> <td>snmp_version</td> <td>StringField</td> <td>snmp version</td> </tr> <tr> <td>snmp_ro_community</td> <td>StringField</td> <td>snmp RO community</td> </tr> <tr> <td>snmp_rw_community</td> <td>StringField</td> <td>snmp RW community</td> </tr> </tbody> </table>	Поле	Тип	Описание	address	StringField	Адрес объекта	port	IntegerField	Порт объекта	profile	StringField	Профиль объекта	user	StringField	CLI user	password	StringField	CLI password	super_password	StringField	enable password	snmp_version	StringField	snmp version	snmp_ro_community	StringField	snmp RO community	snmp_rw_community	StringField	snmp RW community
		Поле	Тип	Описание																												
		address	StringField	Адрес объекта																												
		port	IntegerField	Порт объекта																												
		profile	StringField	Профиль объекта																												
		user	StringField	CLI user																												
		password	StringField	CLI password																												
		super_password	StringField	enable password																												
		snmp_version	StringField	snmp version																												
snmp_ro_community	StringField	snmp RO community																														
snmp_rw_community	StringField	snmp RW community																														
args	DictField	Входные параметры скрипта																														
activator	StringField	Название пула активаторов																														
shard	IntField	ID шарда																														
instance	StringField	Название instance активатора в пуле, который выполняет задачу																														
debug	BooleanField	True, если необходимо создать тушенку																														
caps	DictField	Object capabilities в формате Ключ -> значение																														


Результат работы записывается в GridFS в формате JSON (как получено с активатора). В качестве ключа используется id скрипта

 Куда записывать тушенку?

API для запуска скрипта

```
task = managed_object.scripts.get_interfaces(name="Gi 0/1")
```

Возвращаемая задача имеет тип Future

 Использовать ли механизм long running jobs?

API для запуска группы задач

Аналог старых ReduceTask. Используется при необходимости запуска сразу нескольких задач

```
group = GroupTask()
group.add(managed_object.scripts.get_interfaces(name="Gi 0/1",
group=group) #
group.add(selector, "get_interfaces", name="Gi 0/1") #
task = group.run()
```

Возвращаемая задача имеет тип Future. При запуске задачи создается соответствующая запись в коллекции Scripts. Зап

Возможно использование нескольких вложенных групп:

CURRENT (v. 3) 29.01.2015 08:27

v. 5 29.01.2015 06:42

v. 4 29.01.2015 06:33

v. 3 29.01.2015 06:29

```
group1 = GroupTask()
group2 = GroupTask()
group1.add(group2)
task = group1.run()
```

v. 2

29.01.2015 05:54

Переделка SAE

SAE реализован как HTTP-сервер на базе tornado. В качестве протокола по взаимодействию с активаторами используется JSON-RPC. SAE пассивен, задачи выдаются по запросу активатора. Используется механизм long polling, при котором активаторы, для которых нет задач, зависают на незакрытом коннекте, пока не появятся задачи. Данный механизм позволяет сократить задержку на передачу задания на активатор.

Реализуемые сервисы:

SAService (/SAService/)

v. 1

29.01.2015 05:49

Метод	Входные параметры			Выходные параметры	Описание																	
get_tasks	Имя	Тип	Описание	Список значений, длиной не более n	Активатор получает новые задачи, вызывая функцию get_tasks. Для снижения количества HTTP-запросов активатор может запросить несколько задач сразу. Метод подвисает на long polling. Обрыв по таймауту означает, что на данный момент задач нет и необходимо повторить попытку. Для поиска новой задачи используется функция mongodb findAndModify. Запрос для поиска:																	
	instance	int	instance id																			
	n	int	количество задач, которые активатор готов взять на выполнение	<table border="1"> <thead> <tr> <th>Имя</th> <th>Тип</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>str</td> <td>ID задачи</td> </tr> <tr> <td>script</td> <td>str</td> <td>Имя скрипта</td> </tr> <tr> <td>access_profile</td> <td>dict</td> <td>Содержание access_profile из коллекции scripts</td> </tr> <tr> <td>args</td> <td>dict</td> <td>Входные параметры скрипта</td> </tr> <tr> <td>caps</td> <td>dict</td> <td>Capabilities</td> </tr> </tbody> </table>	Имя	Тип	Описание	id	str	ID задачи	script	str	Имя скрипта	access_profile	dict	Содержание access_profile из коллекции scripts	args	dict	Входные параметры скрипта	caps	dict	Capabilities
Имя	Тип	Описание																				
id	str	ID задачи																				
script	str	Имя скрипта																				
access_profile	dict	Содержание access_profile из коллекции scripts																				
args	dict	Входные параметры скрипта																				
caps	dict	Capabilities																				
					<pre>{ status: "W", activator: <activator name>, shard: <shard id> }</pre> <p>Для модификации данных используется</p> <pre>{ \$set: { status: "R", instance: <activator instance> } }</pre>																	

set_result	Имя	Тип	Описание	boolean True - результат записан False - ошибка записи	Результат выполнения скрипта. Функция вызывается на каждого скрипта по отдельности
	id	str	id задачи		
	result_code	int	Статус завершения . 0 - успешное завершение		
	result	object	Сериализованный результат работы		

!!! throttling

!!! Как учитывать object status?

- Сервис /register/ - дергается активатором для мониторинга
 - POST /register/
 - -> {"expires": seconds}
 - Получив ответ активатор должен повторно дернуть /register/ через seconds/2, в случае ошибки - через seconds/8
- POST /ping/ - дергается NOC-ping с изменениями статуса
 - На входе список
 - timestamp
 - object id
 - status

Переделка активатора

- AutoConf daemon, получает в качестве конфига JSON
 - max_sessions
- http client регистрации, крутится в отдельном потоке, реализует сервис /register/
- threadpool - выполняет script runner
- network -- поток с SocketFactory
- dispatcher - отдельный поток, который по запросу script_runner'a запрашивает новые задачи и пересылает на SAE результат (через сервис /task/)

noc-ping

Задача пингования хостов переключается на отдельный демон noc-ping

- AutoConf daemon, получает в качестве конфига хосты для пропинговки
- SocketFactory с ping socket'ом, которые занимаются пропинговкой
- sender - отсылает изменения статусов объектов на SAE

noc-collector

Удаляемые зависимости

- protocol buffers

Открытые вопросы

Открытые вопросы, не решенные проблемы:

Вопрос	Варианты решения

Related Issues

NOC-291, NOC-1495

Отложенные задачи

Задача	Причина переноса	Примечания