Synnefo - Feature # 367

Status:	New	**Priority:**	Low
Author:	Vangelis Koukis	**Category:**	obsolete_AAI
Created:	04/04/2011	**Assignee:**	Giorgos Verigakis
Updated:	06/29/2011	**Due date:**	
Subject:	Rate-limiting των requests στο API		

Description

Πρέπει να υλοποιηθεί rate limiting των requests στο API, ως anti-DoS μηχανισμός.
Το γράφω εδώ για να υπάρχει ως εκκρεμμότητα.

Related issues:

related to Synnefo - Feature # 405: Απόλυτος περιορισμός σε πλήθος πόρων (quo...	**Closed**	**04/08/2011**

History

#1 - 04/08/2011 10:02 pm - Vangelis Koukis

- Category changed from Cyclades API to obsolete_AAI

#2 - 04/08/2011 10:05 pm - Vangelis Koukis

- Target version set to v0.5

#3 - 04/10/2011 11:09 am - Giorgos Gousios

Δεν νομίζω ότι είναι θέμα μόνο του AAI. Πρέπει να μπει ένα middleware που περιορίζει τα requests ανά IP (-> πριν το αντίστοιχο του AAI) και ίσως μετά κάτι για τον περιορισμό ανά χρήστη.

#4 - 04/10/2011 02:23 pm - Panagiotis Louridas

Παιδιά αυτό είναι overengineering.

(1) DoS μετά από AAI σημαίνει ότι κάποιος τρελός κάνει εξυπνάδες επώνυμα.

(2) DoS πριν από AAI κολλάει στο authentication layer.

#5 - 04/10/2011 08:12 pm - Giorgos Gousios

Παρ'ολα αυτά:

1. Πρέπει να καταγράφεται και να απομονώνεται.
2. Χωρίς περιορισμό μπορεί να κάνει το σύστημα να μην ανταποκρίνεται επαρκώς

Μια άλλη ιδέα θα ήταν να εφαρμόσουμε κάποια τεχνική tarpitting σε επίπεδο TCP (στο router) για να μην επιβαρύνουμε την υλοποίησή μας, τουλάχιστον όσον αφορά το σημείο 2 του Πάνου.

http://en.wikipedia.org/wiki/Tarpit_(networking)

#6 - 04/10/2011 08:36 pm - Panagiotis Louridas

Ποιο είναι το σενάριο για το οποίο μιλάμε; Μπορεί κάποιος να το επεξεργαστεί; Για παράδειγμα το:

- Ο χρήστης ξεκινάει όσες μηχανές μπορεί ταυτόχρονα

δεν είναι πρόβλημα, γιατί δεν θα μπορεί λόγω quota.

Το

- Ο χρήστης ανοιγοκλείνει μηχανές όσο πιο γρήγορα μπορεί, ή δημιουργεί και σβήνει images όσο πιο γρήγορα μπορεί, κ.λπ.

είναι πρόβλημα; Πώς συμπεριφέρεται το Ganeti σε τέτοια διαδοχικά request (π.χ. κλείσιμο VM που δεν έχει προλάβει να ανοίξει);

Φαντάζομαι ότι κάποιο queue κάποια στιγμή θα γεμίζει. Θα είναι το 0mq (ή όποιο άλλο χρησιμοποιήσουμε) αυτό; Μήπως το πρόβλημα λοιπόν θα εντοπίζεται κατευθείαν εκεί, με ένα μήνυμα προς τους administrators όταν τα job queues ξεπερνούν κάποιο threshold;

Με άλλα λόγια: πριν ψάξουμε για λύσεις ας δούμε το πρόβλημα.

#7 - 04/10/2011 08:52 pm - Vangelis Koukis

Πράγματι, το να ξεκινήσει υπερβολικά μεγάλο αριθμό μηχανών ο χρήστης δεν είναι πρόβλημα.
Αλλά το θέμα με το ρυθμό υποβολής και τελικά των αριθμό των requests που είναι outstanding είναι πρόβλημα.

Ο χρήστης μπορεί να στείλει μια εντολή να ανάψει ένα μηχάνημα, μετά μπορεί να στείλει εντολή να σβήσει, μετά να ανάψει και πάει λέγοντας. Κάθε εντολή υποβάλλεται στην ουρά του Ganeti, παίρνει το νούμερό της και κάποια στιγμή ολοκληρώνεται. Το Ganeti απλώς δέχεται δουλειές στην ουρά του.

Είναι λογικό οι διαχειριστές να παρακολουθούν ίσως με αυτόματα εργαλεία την κατάσταση του Ganeti και της ουράς του, αλλά αυτό δεν σημαίνει ότι το frontend (το Synnefo) δεν πρέπει να έχει κάποιον στοιχειώδη μηχανισμό περιορισμού του ρυθμού υποβολής δουλειών από το χρήστη. Αλλιώς, ένας μεμονωμένος χρήστης μπορεί να προκαλέσει μεγάλο πρόβλημα, στέλνοντας με όσο μεγαλύτερο ρυθμό μπορεί και τιγκάροντας με δουλειές του στυλ "άναψε το Α", "σβήσε το Α", την ουρά.

Οπότε, ψηφίζω να υλοποιηθεί rate limiting στο API, όπως εξάλλου προβλέπεται από το spec, το οποίο προσδιορίζει και τρόπο με τον οποίο ο χρήστης του API μπορεί να μάθει ποια είναι τα μέγιστα rates.

#8 - 04/10/2011 09:10 pm - Panagiotis Louridas

Αυτό που λες είναι limiting σε επίπεδο HTTP requests (HTTP request rate limiting), του τύπου που εφαρμόζει το twitter, foursquare, κ.λπ. Οπότε το ζήτημα είναι να δούμε πώς υλοποιούνται τέτοια σχήματα.

Οπότε επιμένω σε αυτό που είπα:

- Να δει κάποιος ποιο ακριβώς είναι το πρόβλημα (είναι τελικά HTTP request rate limiting;) και ποιες είναι ενδεχόμενες λύσεις (σε επίπεδο Django, πιο πάνω, ή πιο κάτω);

- Να εκτιμηθεί πόσο χρόνο χρειάζεται η υλοποίησή του, και αν χωράει στο 0.5.

Αλλιώς εύκολα το 0.5 θα παραδωθεί το Δεκέμβρη.

#9 - 04/10/2011 09:29 pm - Vangelis Koukis

Μα δεν νομίζω ότι διαφωνούμε σε αυτό!

Συμφωνώ απόλυτα ότι πρόβλημα πρέπει να περιγραφεί με ακρίβεια από όποιον αναλάβει το ticket. Πάντως, σε πρώτη σκέψη, νομίζω ότι η λύση πρέπει να είναι στο επίπεδο του API, όχι πιο πάνω ή πιο κάτω από το Django, γιατί: αν είναι πιο πάνω πώς θα καταγράφεται **ποιος** χρήστης είναι αυτός που χτυπάει το request limit; αν είναι πιο κάτω, πού θα είναι; στην ουρά του Ganeti, στη ΒΔ [που μπορώ να χτυπάω ανηλεώς κάνοντας GET /servers, πχ]; παρακολούθηση θα υπάρχει εκεί, αλλά νομίζω δεν είναι σωστό να μπορεί ο χρήστης να σηκώσει φόρτο σε τόσο χαμηλά [και κρίσιμα] επίπεδα του συστήματος χωρίς κάτι να κάνει κάποιου είδους rate limiting πριν. Επίσης, δεν είναι όλα τα requests ίδια. Άλλο το να υποβάλλω δουλειά στο Ganeti, άλλο το να κάνω GET στη βάση. Το spec προβλέπει διαφορετικά όρια για διαφορετικά είδη requests, νομίζω ότι είναι λογικό. Αν ο περιορισμός γίνει υπερβολικά ψηλά, μπαίνουν όλα [και οι χρήστες και το είδος των requests] στο ίδιο τσουβάλι.

Όσον αφορά την εκτίμηση του απαιτούμενου χρόνου για υλοποίηση, έχεις δίκιο πρέπει να δούμε αν χωράει. Έχεις επίσης απόλυτο δίκιο, αν αρχίσουμε να βάζουμε features όπως μας έρχονται, άνετα η v0.5 θα παραδοθεί από τον Αγ. Βασίλη.
Αλλά για να είμαι ειλικρινής θεωρώ ότι η μελέτη και υλοποίηση ενός τέτοιου μηχανισμού είναι κρίσιμη για το σύστημα, χρειάζεται σκέψη και πρέπει να είναι το feature που θα φύγει τελευταίο, (ότ)αν δούμε ότι δεν χωράνε όλα.

#10 - 04/10/2011 09:57 pm - Panagiotis Louridas

Το API περιγράφει το πώς παίρνεις τα limits, όχι πώς υλοποιούνται.

- Μήπως όταν φτάσουν 5000 POST requests / min στην εφαρμογή από έναν χρήστη είναι ήδη αργά;

Η απάντηση για μένα είναι ότι δεν έχω ιδέα. Πολύ θα χαρώ να μάθω λοιπόν, όπως και πόσο χρόνο θα χρειαστεί για να μάθω και να λυθεί το πρόβλημα :-)

#11 - 04/10/2011 10:00 pm - Panagiotis Louridas

Και να υπενθυμίσω γιατί είναι χαμηλά στις δικές μου προτεραιότητες:

- Γιατί το ζώον που θα δώσει 5000 requests / min θα είναι ακριβώς ζώον, γιατί αμέσως θα ξέρουμε ποιος είναι.

Ιδιοφυίες σπανίζουν, αλλά το ίδιο σπάνιοι είναι και οι βλάκες.

#12 - 04/10/2011 10:28 pm - Giorgos Gousios

Δεν θα ξέρουμε ποιο είναι το ζώον αν δεν έχουμε βάλει ένα καμπανάκι να χτυπάει όταν κάποιος ξεπεράσει τα 5000 req/sec. Γι αυτό και επιμένω ότι τουλάχιστον monitoring μπορούμε/πρέπει να έχουμε. Επίσης, δεν νομίζω ότι το να γεμίσει μια ουρά και να αρχίσει να χάνει δουλειές εξ' αιτίας κάποιου ζώου είναι λύση στο πρόβλημά μας. Ας το συζητήσουμε από κοντά την Τρίτη, αφού ψάξουμε όλοι καλύτερα τι επιπτώσεις μπορούν να έχουν τα ζώα στον ωκεανό :-)

#13 - 04/11/2011 08:14 am - Panagiotis Louridas

Το πού βοσκάνε τα ζώα το βρίσκεις από τα logs.

Εγώ προτείνω για κάθε ρίσκο που εντοπίζουμε (γιατί δεν θα είναι το μόνο), να δουλεύουμε όπως είθισται στη διαχείριση έργων:

- Για τη διαχείριση του ρίσκου λαμβάνουμε υπόψη την πιθανότητα του ενδεχόμενου επί το κόστος του, **όχι μόνο το κόστος**.

Αλλιώς το έργο απλώς υπερβαίνει και χρόνο και κόστος.

#14 - 04/11/2011 08:40 am - Panagiotis Louridas

Και για να είμαι πιο συγκεκριμένος, προς το παρόν θα έβαζα ένα rate limit σε επίπεδο iptables (θέτοντας π.χ. max 50 connections σε 60 sec από ένα IP address) και όταν έχουμε χρόνο κοιτάζουμε πιο ολοκληρωμένη λύση (αν στο μεταξύ έχει ποτέ εμφανιστεί πρόβλημα).

Χάνω κάτι;

#15 - 04/11/2011 09:49 am - Giorgos Gousios

Σε αυτό συμφωνώ, αυτό πρότεινα και στην πρώτη μου απάντηση για τους μη εγγεγραμένους χρήστες. Αυτό που πρότεινα επιπλέον, και δεν νομίζω ότι έχει μεγάλο κόστος ή ρίσκο, είναι να καταγράφουμε το ρυθμό αιτήσεων από εγγεγραμένους χρήστες. Αφού ούτως η άλλως καταγράφουμε τις αιτήσεις, το να καταγράφουμε και το ρυθμό τους δεν έχει μεγάλη διαφορά στην υλοποίηση. Τώρα όσον αφορά τη συμμόρφωση της υλοποίησής μας με το API, αυτό είναι κάτι που πρέπει να δούμε. Για αρχή, μπορούμε απλά να επιστρέφουμε στατικές τιμές για τα limits. Αργότερα, μπορούμε να ρωτάμε το iptables κατευθείαν.

#16 - 04/14/2011 02:56 pm - Giorgos Verigakis

Πού καταλήγουμε τελικά με αυτό; Θα κάνει το API accounting των requests των χρηστών; Και αν ναι, πόσο fine grained θα είναι αυτή η καταγραφή; Έχουμε διαφόρων ειδών API functions (listings, creations, deletions, actions, server-related, image-related, κτλ), θα καταγράφονται όλα ανεξάρτητα ή θα υπάρχει κάποιου είδους ομαδοποίηση;

#17 - 04/14/2011 03:20 pm - Vangelis Koukis

Δεν έχουμε καταλήξει κάπου ακόμη. Δεδομένου ότι έχουμε πάρα πολλά να κάνουμε για το v0.3, ας συνεχίσουμε να το σκεφτόματε χωρίς να γίνει προτεραιότητα. Αν έχει ωριμάσει η συζήτηση, μπορεί να μπει για v0.4, αλλά προς το παρόν νομίζω μπορείς να το αφήσεις.

#18 - 06/29/2011 01:40 pm - Vangelis Koukis

Σε προηγούμενη συνάντηση αποφασίστηκε ότι αυτό το feature δεν θα είναι μέρος της v0.5, αλλά θα υλοποιηθεί μετά από αυτή.
Έχει υλοποιηθεί λειτουργικότητα για quota στο πλήθος των εικονικών μηχανών στο #703.

#19 - 06/29/2011 01:40 pm - Vangelis Koukis

- Target version deleted (v0.5)