

Ενσωμάτωση μηχανισμών caching,  
observability και self-healing στην  
υποδομή διαχείρισης δεδομένων IoT της  
πλατφόρμας Nostradamus

Δημήτριος Ντέντας

5 Απριλίου 2025

# Περιεχόμενα

|   |                                     |   |
|---|-------------------------------------|---|
| 1 | Επισκόπηση της Ερευνητικής Περιοχής | 1 |
|   | Βιβλιογραφία                        | 4 |

# 1

## Επισκόπηση της Ερευνητικής Περιοχής

Στη σύγχρονη γεωργία, ένας δυσλειτουργικός αισθητήρας που περνά απαρατήρητος για ώρες μπορεί να οδηγήσει σε αποτυχημένο πότισμα ή καταστροφή καλλιεργειών. Αυτό το ενιαίο σημείο αποτυχίας αναδεικνύει την ανάγκη για αυτόνομα, ανθεκτικά και παρατηρήσιμα συστήματα σε κλίμακα. Η ανάγκη αυτή εναρμονίζεται με τον οδικό χάρτη προς ανθεκτικά Κυβερνοφυσικά Συστήματα (CPS) που περιγράφεται από τους Ratasich et al. [1], οι οποίοι δίνουν έμφαση στην ανίχνευση ανωμαλιών κατά τη διάρκεια λειτουργίας, στην απομόνωση σφαλμάτων και στην αυτοϊαση σε δυναμικά Internet of Things (IoT) περιβάλλοντα.

Η αυτονομία, η ανθεκτικότητα, αλλά και η παρατηρησιμότητα σε μεγάλης κλίμακας real time συστήματα αποτελούν βασικές απαιτήσεις για την υποστήριξη κρίσιμων εφαρμογών, ειδικά στον τομέα του Internet of Things (IoT). Καθώς η ποσότητα των παραγόμενων δεδομένων αυξάνεται εκθετικά, οι υποδομές που επεξεργάζονται και αποθηκεύουν αυτά τα δεδομένα οφείλουν να είναι όχι μόνο αποδοτικές, αλλά και αυτοπροσαρμοζόμενες, με δυνατότητα επιτήρησης και γρήγορης αποκατάστασης σε περιπτώσεις σφαλμάτων ή αστοχιών. Οι σύγχρονες cloud-native τεχνολογίες παρέχουν τα εργαλεία και τα πρότυπα για την υλοποίηση τέτοιων χαρακτηριστικών.

Παρότι υπάρχουν επιμέρους τεχνολογίες που προσφέρουν caching, observability ή self-healing δυνατότητες, η ενσωμάτωσή τους σε πραγ-

---

ματικές πολυεπίπεδες IoT πλατφόρμες συνοδεύεται από σημαντικές προκλήσεις. Στα κατανεμημένα συστήματα, η προσθήκη μιας νέας υπηρεσίας ή λειτουργικότητας συνοδεύεται συχνά από μη προβλέψιμες επιπτώσεις στην απόδοση του συστήματος. Αν και οι μηχανισμοί όπως το caching και το observability αποσκοπούν στην βελτίωση της εμπειρίας και της διαχειρισιμότητας, μπορούν, ως παράπλευρη συνέπεια, να οδηγήσουν σε αύξηση του latency ή της κατανάλωσης πόρων. Το φαινόμενο αυτό αποδίδεται στην πολυπλοκότητα των αλληλεπιδράσεων μεταξύ μικροϋπηρεσιών και την έλλειψη πλήρους visibility σε runtime επίπεδο [2].

Η ανάγκη για αποδοτική ροή δεδομένων μεταξύ μικροϋπηρεσιών οδηγεί συνήθως στην υιοθέτηση τεχνολογιών όπως το *Apache Kafka*, ένα - πλέον - *de facto* πρότυπο για την υλοποίηση ανθεκτικών και επεκτάσιμων messaging υποδομών, χάρη στη δυνατότητά του να αποθηκεύει τα μηνύματα σε μορφή καταγραφής (log-based) [3].

Η αποθήκευση μεγάλου όγκου δεδομένων σε κατανεμημένα συστήματα βασίζεται - κυρίως - σε βάσεις δεδομένων τύπου *wide-column*, με το *Apache Cassandra* να αποτελεί ένα από τα πιο διαδεδομένα και ώριμα συστήματα σε αυτόν τον χώρο. Το *Cassandra* ακολουθεί το μοντέλο *eventual consistency*, υποστηρίζει κατανεμημένη αποθήκευση με replication και partitioning, και έχει σχεδιαστεί για *write-heavy* εφαρμογές [4].

Σε ότι αφορά την ανάγνωση των αποθηκευμένων δεδομένων αυτού του όγκου, αλλά και δεδομένης της συνεχόμενης άφιξης τους, τα προσωρινά δεδομένα σε μνήμη (*in-memory caching*) αποτελούν κρίσιμο μηχανισμό βελτιστοποίησης της απόδοσης. Δύο από τις πιο διαδεδομένες τεχνολογίες στον τομέα αυτό είναι τα συστήματα *Memcached* [5] και *Redis* [6], τα οποία υλοποιούν μηχανισμούς προσωρινής αποθήκευσης ζευγών κλειδιού-τιμής, με στόχο τη μείωση του χρόνου και του κόστους προσπέλασης σε βάσεις δεδομένων. Η βασική διαφορά τους έγκειται κυρίως στην αρχιτεκτονική του ολικού συστήματος, καθώς και στον τρόπο διαχείρισης της ταυτόχρονης εκτέλεσης - με χρήση *multithreading* (*Memcached*) ή *event loop* (*Redis*).

Η ανάγκη για παρατηρησιμότητα, αποδοτική διαχείριση προσωρινών δεδομένων και σταθερή ροή μηνυμάτων σε περιβάλλοντα με κατανεμημένη υπολογιστική λογική, καθιστά την πλατφόρμα *Kubernetes* - συχνά - απαραίτητο δομικό στοιχείο. Η χρήση της επιτρέπει την υλοποίηση μηχανισμών αυτόματης αποκατάστασης, *autoscaling*, και επιτήρησης σε επίπεδο υπηρεσίας. Σε συνδυασμό με συστήματα όπως το *Prometheus* (και τα συνεργατικά του υποσυστήματα *Grafana* και *AlertManager*) [7] - κοινώς τη *lingua franca* του *observability* - συμβάλλει στον σχεδιασμό και στην υλοποίηση μεγάλων υποδομών, πληρώνοντας τα κριτήρια μιας

---

παραγωγικής (production-grade) αρχιτεκτονικής.

Κατ' επέκταση, η ερευνητική συνεισφορά εστιάζει όχι μόνο στην επιλογή επιμέρους τεχνολογιών, αλλά κυρίως στη συνεκτική και δυναμική ενορχήστρωσή τους, με στόχο την επίτευξη αυτονομίας και διαχειρισιμότητας σε cloud-native κατανεμημένα περιβάλλοντα υψηλής πολυπλοκότητας.

# Βιβλιογραφία

- [1] Denise Ratasich, Faiq Khalid, Florian Geissler, Radu Grosu, Muhammad Shafique, and Ezio Bartocci. A roadmap toward the resilient internet of things for cyber-physical systems. *IEEE Access*, 2019.
- [2] Martin Kleppmann. *Designing Data-Intensive Applications*. O'Reilly Media, 2017.
- [3] Rishika Shree, Tanupriya Choudhury, Subhash Chand Gupta, and Praveen Kumar. Kafka: The modern platform for data management and analysis in big data domain. In *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, 2017.
- [4] Avinash Lakshman and Prashant Malik. Cassandra: a decentralized structured storage system. *SIGOPS Oper. Syst. Rev.*, 2010.
- [5] Rajesh Nishtala, Hans Fugal, Steven Grimm, Marc Kwiatkowski, Herman Lee, Harry C. Li, Ryan McElroy, Mike Paleczny, Daniel Peek, Paul Saab, David Stafford, Tony Tung, and Venkateshwaran Venkataramani. Scaling memcache at facebook. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2013.
- [6] Josiah Carlson. *Redis in Action*. Manning Publications, 2013.
- [7] Sneha M. Pragathi B.C., Hrithik Maddirala. Implementing an effective infrastructure monitoring solution with prometheus and grafana. *International Journal of Computer Applications*, 2024.