Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Τεχνολογία Λογισμικού

8ο Εξάμηνο 2022-23

Αρχιτεκτονική λογισμικού (1 / 2)

Δρ. Κώστας Σαΐδης (saiko@di.uoa.gr)

Αρχιτεκτονική λογισμικού

Περί τίνος πρόκειται

Η λήψη των θεμελιωδών δομικών και σχεδιαστικών αποφάσεων για το λογισμικό που είναι ακριβό (σε κόπο, χρόνο, χρήμα) να αλλάξουν άπαξ και υλοποιηθούν.

Ποιοτικά χαρακτηριστικά αρχιτεκτονικής κατανεμημένων συστημάτων

Ορισμός

Ένα κατανεμημένο σύστημα αποτελείται από ξεχωριστά συστατικά που:

- Λειτουργούν σε ένα δίκτυο υπολογιστών.
- Επικοινωνούν μεταξύ τους μέσω ανταλλαγής μηνυμάτων.
- Αλληλοεπιδρούν για την επίτευξη ενός κοινού στόχου.

Πλάνες σχετικά με τα κατανεμημένα συστήματα (α)

1. Το δίκτυο είναι αξιόπιστο (reliable).
2. Η καθυστέρηση (latency) του δικτύου είναι μηδενική.
3. Το εύρος ζώνης (bandwidth) είναι άπειρο.
4. Το δίκτυο είναι ασφαλές (secure).

Πλάνες σχετικά με τα κατανεμημένα συστήματα (β)

5. Η τοπολογία (topology) του δικτύου δεν αλλάζει.
6. Υπάρχει μόνο ένας διαχειριστής (administrator).
7. Το κόστος μεταφοράς (transport) δεδομένων είναι μηδενικό.
8. Το δίκτυο είναι ομογενές (homogeneous).

Βασικά χαρακτηριστικά ενός κατανεμημένου συστήματος

- Συνέπεια δεδομένων (Consistency)
- Διαθεσιμότητα συστήματος (Availability)
- Αστοχία δικτύου (Network partition)
- Καθυστέρηση αίτησης/απόκρισης (Latency)
- Αιτήσεις ανά μονάδα χρόνου (Throughput)
- Κλιμάκωση (Scalability)

Consistency (C)

- Η συνέπεια των δεδομένων.
- Ζητούμενο: κάθε ανάγνωση (read) λαμβάνει την πιο πρόσφατη ενημέρωση (write) ή το σχετικό σφάλμα.

Προσοχή

Η συνέπεια που υπόσχεται η αρχή ACID των δοσοληψιών στις βάσεις δεδομένων είναι πιο αυστηρή.

ACID Transactions

- Atomicity
  - Η δοσοληψία πετυχαίνει ή αποτυγχάνει πλήρως
- Consistency
  - Μετάβαση της βάσης σε πάντα έκγυρη κατάσταση
- Isolation
  - Απομόνωση της εκτέλεσης των δοσοληψιών
- Durability
  - Μονιμότητα των αποτελεσμάτων των δοσοληψιών

Availability (A)

- Η διαθεσιμότητα της εφαρμογής.
- Ζητούμενο: κάθε αίτηση (request) να λαμβάνει μια απάντηση (μη λάθους).
- Χωρίς να προσφέρονται πάντα εγγυήσεις ότι η απάντηση περιέχει την πιο πρόσφατη ενημέρωση (write).
- Η υψηλή διαθεσιμότητα απαιτεί αντιγραφές (replication).

Network Partition (P)

- Αστοχία δικτύου.
- Επιμερισμός του συστήματος σε "αποσυδεδεμένες" νησίδες.
- Παράδειγμα: απώλεια σύνδεσης με τη βάση δεδομένων.

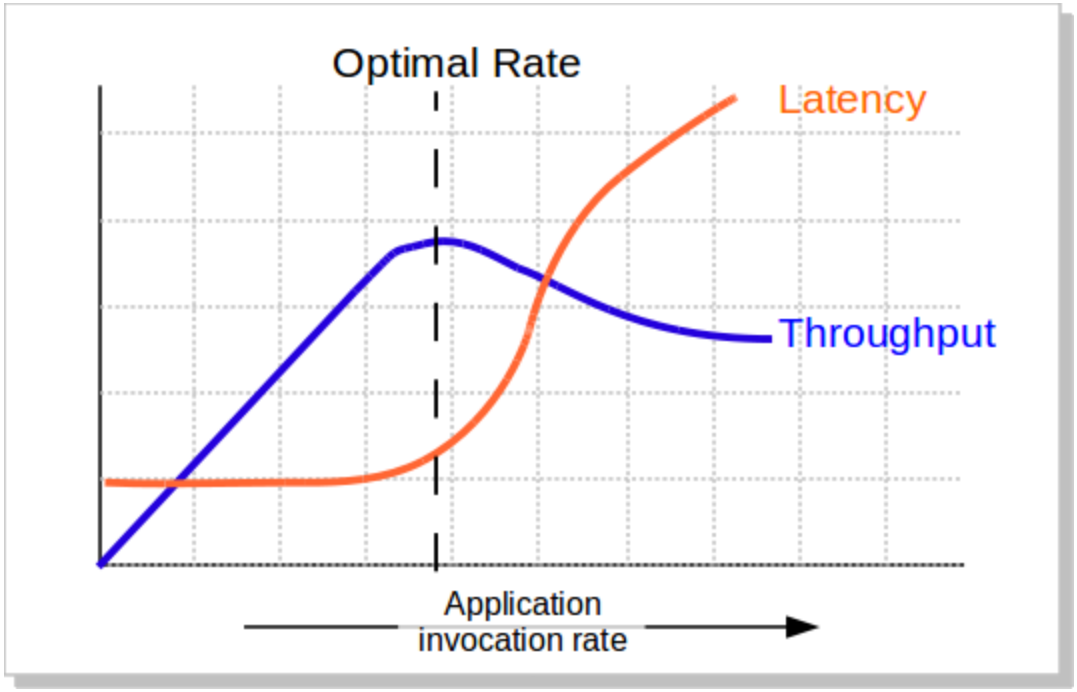
Latency (L)

- Η καθυστέρηση στην απόκριση του συστήματος.
- Ζητούμενο: να ελαχιστοποιηθεί ο χρόνος που απαιτείται για την ικανοποίηση μιας αίτησης.

Throughput

- Το πλήθος των αιτήσεων που ικανοποιούνται από το σύστημα ανά χρονική στιγμή.
- Ζητούμενο: να μεγιστοποιηθεί το πλήθος των αιτήσεων που μπορούν να ικανοποιηθούν ανά χρονική στιγμή.

Latency vs Throughput



docs.voltdb.com

Το θεώρημα CAP

Σε περίπτωση αστοχίας δικτύου (P), θα έχουμε είτε συνέπεια των δεδομένων (C) είτε διαθεσιμότητα της εφαρμογής (A), όχι και τα δύο.

```
if (P) { A or C }
```

Το θεώρημα PACELC

Επέκταση του CAP

Αν δεν υπάρχει P, θα έχουμε είτε συνέπεια των δεδομένων (C) είτε την ελάχιστη δυνατή καθυστέρηση (L), αλλά όχι και τα δύο.

```
if (P) { A or C }
else   { L or C }
```

Λίστα αναγνωσμάτων

Daniel Abadi, "Consistency Tradeoffs in Modern Distributed Database System Design", IEEE Computer, Volume 45, Issue 2, Feb. 2012.

Κατηγοριοποίηση κατανεμημένων συστημάτων

PA/EL

```
if P then A else L
```

PC/EC

```
if P then C else C //ACID databases
```

PC/EL

```
if P then C else L
```

Scalability

Η δυνατότητα ενός συστήματος να επαυξηθεί για να διαχειριστεί αυξημένο φόρτο.

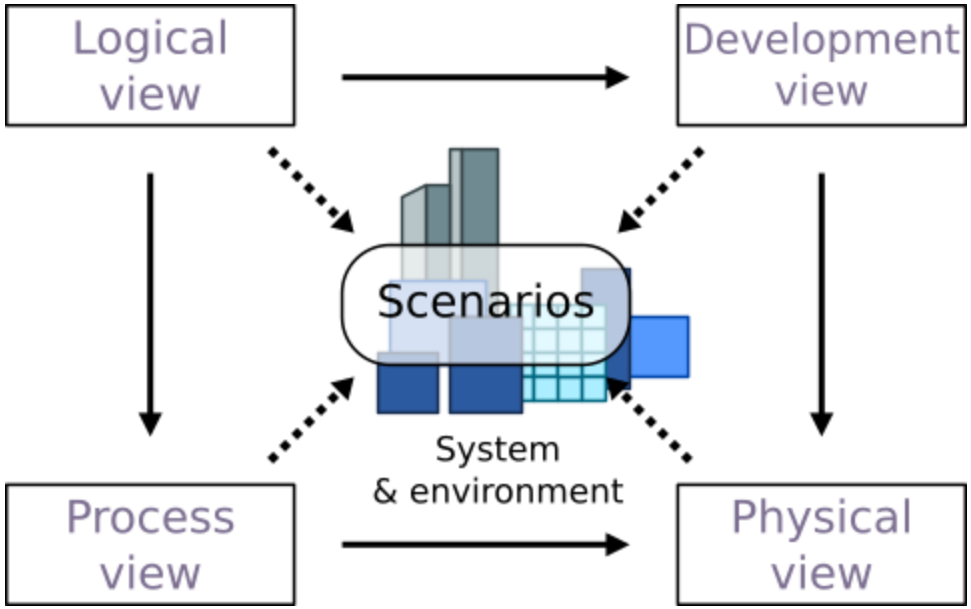
Ειδικότερα

Αν η αύξηση της απόδοσης του συστήματος είναι αναλογική της αύξησης σε υπολογιστικούς πόρους (προσθήκη υλικού), τότε το σύστημα κλιμακώνεται (scales).

Οριζόντια και κάθετη κλιμάκωση

- Οριζόντια (scale out/in): αύξηση/μείωση των κόμβων.
- Κάθετη (scale up/down): αύξηση/μείωση των πόρων ενός κόμβου.

Πολλές αρχιτεκτονικές οπτικές (4+1)



By mpan - Based on File:4+1 Architectural View Model.jpg by User:Mdd, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=50144028

Ειδικότερα

- Logical view: έμφαση στη λειτουργικότητα του συστήματος σε υψηλό επίπεδο
- Physical view: έμφαση στην τοπολογία και διασύνδεση των συστατικών του σε φυσικό επίπεδο (deployment)
- Development view: έμφαση στην οπτική του προγραμματιστή
- Process view: έμφαση στη δυναμική συμπεριφορά του συστήματος κατά την εκτέλεσή του (απόδοση, κλιμάκωση, κτλ.)
- Scenarios - Use case view: έμφαση στη χρηστική πλευρά του συστήματος και στους σχετικούς ελέγχους αποδοχής

Αρχιτεκτονικά πρότυπα (architectural patterns)

Γενικές κι επαναχρησιμοποιήσιμες λύσεις σε κοινά προβλήματα αρχιτεκτονικής.

Αρχιτεκτονικά στυλ (architectural styles)

Όπως και στην "κανονική" αρχιτεκτονική, το στυλ είναι μια συγκεκριμένη μέθοδος κατασκευής που χαρακτηρίζεται από συγκεκριμένα αξιοπρόσεκτα χαρακτηριστικά.

An architectural style is a named collection of architectural design decisions that:

1. are applicable in a given development context,
2. constrain architectural design decisions that are specific to a particular system within that context, and
3. elicit beneficial qualities in each resulting system.

Λίστα αναγνωσμάτων

Richard N. Taylor, Nenad Medvidovic, Eric Dashofy, "Software Architecture: Foundations, Theory, and Practice", 2009, Wiley and Sons, ISBN: 0470167742

Αρχιτεκτονικά στυλ και πρότυπα

- Στο μάθημα δεν θα επιμείνουμε στη διάκριση.
- Είτε τα ονομάσουμε στυλ, είτε πρότυπα, μας εφοδιάζουν με μια κοινή γλώσσα ή λεξιλόγιο για να περιγράψουμε κατηγορίες συστημάτων.
- Είναι σύνηθες να συνυπάρχουν και να συνδυάζονται πολλά αρχιτεκτονικά πρότυπα και στυλ σε μια εφαρμογή.

Στο μάθημα καλύπτουμε (αλφαβητικά)

- Client-Server
- Component-based
- Event-Driven
- Layered / N-tier
- Master-slave/Master-replica
- Message-driven/Publish-subscribe
- Microservices

- Model-View-Controller (MVC)
- Model-View-ViewModel (MVVM)
- Peer-to-peer (P2P)
- Pipeline / Pipe-filter
- Representation State Transfer (REST)
- Service-oriented
- Share-nothing
- World Wide Web