

# Open4Tech Guest Lecture

## Chasing Nodes...

**Victor Ciura**

Principal Engineer, CAPHYON

[www.caphyon.com](http://www.caphyon.com)

Durata: **120 minute**

Dificultate: **medie**

Stil: **interactiv** (prezentare de exemple pe care să le analizăm prin dialog Q&A cu studenții)

Mediu: **slides, live code demos**

Homework: **research topics** for students (implementing proposed alternative solutions)

Prerequisites: **C++ STL data structures & algorithms, node-based data structures**

### Despre conținutul tehnic:

Am colectat și pregătit exemple utile și interesante de contrastare între **node-based data structures** și **hash tables** în practica noastră, din codul **Advanced Installer (Caphyon)**. Am încercat pe cât posibil să adaptăm exemplele extrase pentru a putea fi ușor analizate **izolat** din contextul lor de aplicare.

Ne vom concentra pe exemple ce reprezintă pattern-uri frecvent necesare (utile) nouă. De asemenea, vom prezenta și câteva situații despre care am constatat, de-a lungul timpului, că generează probleme și confuzie pentru programatorii neexperimentați, precum și greșeli comune pe care le putem face atunci când nu suntem atenți.

### Plan de desfășurare:

Începem printr-o scurtă introducere teoretică despre STL **node-based data structures** și **hashed containers**. Nu vom insista pe o prezentare detaliată a fundamentelor teoretice necesare, ci în a explica **de ce** este important să cunoaștem instrumentele puse la dispoziția noastră de biblioteca STL și cu ce ne ajută în practica reală de zi cu zi, în afara exemplurilor canonice / didactice.

Studenții vor putea să descopere, de asemenea, că stăpânirea containerelor hashed și a conceptelor STL (algoritmi), în general, îi va ajuta să economisească timp prețios în elaborarea și testarea soluțiilor pentru probleme de care se vor izbi în viitor.

Restul timpului alocat va fi folosit exclusiv pentru **exemple** de *lookup strategies* și discuții pe marginea acestora. Bucățile de cod vor fi scurte și bine izolate astfel încât să nu necesite informație suplimentară despre contextul de aplicare în scenariul real.

Dorim ca prezentarea să aibă un **stil interactiv**, să întreținem un dialog cu studenții, adresând întrebări legate de fiecare exemplu prezentat. Întrebările nu vor fi în stil quizz, ci mai degrabă de **opinie**, de genul:

- “Ce credeți că face codul de pe slide?”
- “Ce potențiale probleme are codul acesta?”
- “Cum putem îmbunătăți / corecta / simplifica codul?”

Încurajăm studenții să pună întrebări legate de exemplele prezentate, atunci când consideră că un concept a fost neclar sau dacă nu înțeleg anumite părți din cod.

Exemplele vor acoperi, neexclusiv, următoarele **concepte**, pe care le considerăm fundamentale:

- *Trees and other node-based data structures*
- *STL associative data structures (set/map)*
- *STL hashed containers*
- *desirable properties of a good hashing function/algorithm*
- *custom extension points for STL containers*
- *efficiency vs. performance*
- *and (of course) their applications in industrial software*

### Prerequisites:

- Familiaritate cu utilizarea hash containers STL (std::unordered\_set, std::unordered\_map)
- Familiaritate cu utilizarea associative containers STL (std::map)
- Familiaritate cu concepte de baza legate de STL algorithms & data structures

### Link-uri utile:

- <https://en.cppreference.com/w/cpp/container/map>
- <https://en.cppreference.com/w/cpp/container/set>
- [http://en.cppreference.com/w/cpp/container/unordered\\_map](http://en.cppreference.com/w/cpp/container/unordered_map)
- <http://en.cppreference.com/w/cpp/utility/hash>
- <https://en.cppreference.com/w/cpp/algorithm>