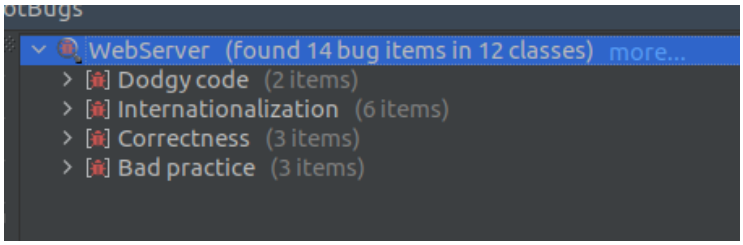
Dobre Ewald-Emanuel, AN IV, CTI-RO

I) Analiza statica

La o prima rulare a SpotBugs gasim 14 potentiale bug-uri.



1. Primul bug gasit se refera la scrierea intr-un camp static in cadrul unei metode nestatice:

```
public class ConfigurationManager {

    private static ConfigurationManager configurationManager;
    private static Configuration currentConfig;

    public void loadConfigurationFile(String filePath) throws ConfigurationFileException {
        ...

        currentConfig = new Configuration(Integer.parseInt(lines.get(0)), lines.get(1), lines.get(2),
lines.get(3), lines.get(4), lines.get(5));

    }

    ...

}
```

Acesta nu este un bug… avem o clasa formata dupa pattern-ul Singleton (ConfigurationManager) care contine un camp static currentConfig. Nu este o problema ca scriem intr-un camp static intr-o metoda nestatica pentru ca exista un singur obiect pe care se poate apela metoda. Ar exista o problema intr-adever daca de pe diverse thread-uri s-ar apela metoda loadConfigurationFile()… caz in care functia ar trebui facuta mai robusta.

2. Un set de bug-uri gasite sunt legate de folosirea System.exit(), care opreste intregul JVM cand e apelata. Am folosit aceasta functie in cazul in care apare o exceptie legata de initializarea server socket-ului sau de citirea din socket. Intr-adevar, pentru o functionare fara oprire a server-ului ar trebui tratate exceptiile in mod diferit.

3. Alta eroare e legata de neverificarea numarului de bytes cititi in cadrul unui InputStream(sa fie egali numarul de bytes cititi cu numarul de bytes ceruti). Acesta este intr-adevar un bug.

4. Urmatoarea eroare indicata se regaseste in mai multe locuri:

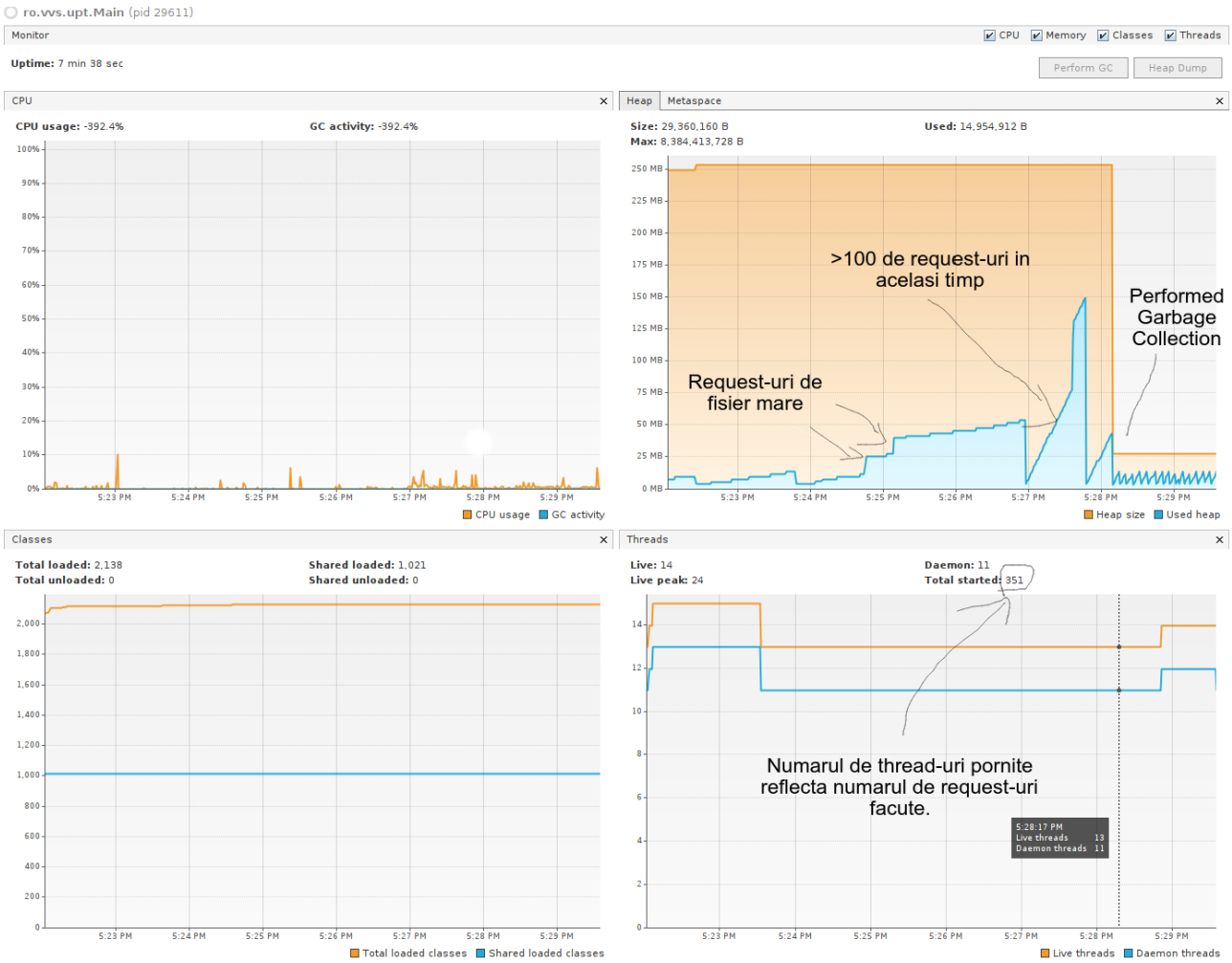
```
Scanner in = new Scanner(System.in);
PrintWriter out = new PrintWriter(s.getOutputStream());
in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
```

SpotBugs atentioneaza ca pe diferite platforme encoding-ul poate fi diferit si, fiind facuta o conversie intre bytes si String pot aparea comportamente diferite in functie de platforma.

5. Restul potentialelor bug-uri descoperite sunt legate de lipsa unor null check-uri.

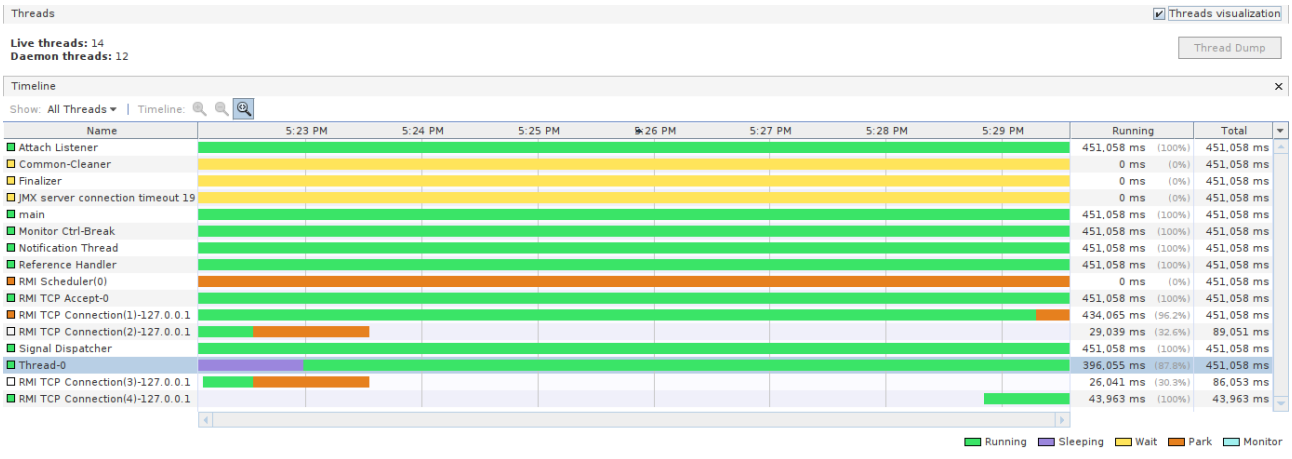
Dobre Ewald-Emanuel, AN IV, CTI-RO

II) Analiza dinamica



Am rulat server-ul si am navigat putin prin paginile html, dar acestea nu au avut un impact vizibil. Dupa adaugarea unui fisier de 10MB+ impactul se vede clar in imagine. Dar am vrut mai mult, asa ca am produs 100+ request-uri artificiale in acelasi timp. La final am utilizat functionalitatea de GC din JvisualVM. De asemenea, in sectiunea Threads se putea vedea ca atunci cand avea loc un request, se incrementa numarul de “Total started”.

Dobre Ewald-Emanuel, AN IV, CTI-RO



In imaginea de mai sus se observa Thread-0 care e thread-ul principal. Fiindca server-ul e initial in starea Stopped, thread-ul nu face nimic(e in starea “Sleeping”). Apoi, intra in starea Running, in care ramane pana la sfarsit.
Se mai observa niste thread-uri care incep cu “RMI”. Acestea sunt necesare pt JvisualVM, prin ele sunt transmise datele despre thread-uri, memorie de la programul ce ruleaza. Tot datorita acestor thread-uri apare noise-ul din imaginea cu Heap.