UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE
MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

Lucrare de licență

ATACURI SPECULATIVE

Absolvent
Radu Ștefan-Octavian

Coordonator științific
Titlul și numele profesorului coordonatorului

București, iunie 2021

Rezumat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vitae eros sit amet sem ornare varius. Duis eget felis eget risus posuere luctus. Integer odio metus, eleifend at nunc vitae, rutrum fermentum leo. Quisque rutrum vitae risus nec porta. Nunc eu orci euismod, ornare risus at, accumsan augue. Ut tincidunt pharetra convallis. Maecenas ut pretium ex. Morbi tellus dui, viverra quis augue at, tincidunt hendrerit orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam quis sollicitudin nunc. Sed sollicitudin purus dapibus mi fringilla, nec tincidunt nunc eleifend. Nam ut molestie erat. Integer eros dolor, viverra quis massa at, auctor.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vitae eros sit amet sem ornare varius. Duis eget felis eget risus posuere luctus. Integer odio metus, eleifend at nunc vitae, rutrum fermentum leo. Quisque rutrum vitae risus nec porta. Nunc eu orci euismod, ornare risus at, accumsan augue. Ut tincidunt pharetra convallis. Maecenas ut pretium ex. Morbi tellus dui, viverra quis augue at, tincidunt hendrerit orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam quis sollicitudin nunc. Sed sollicitudin purus dapibus mi fringilla, nec tincidunt nunc eleifend. Nam ut molestie erat. Integer eros dolor, viverra quis massa at, auctor.

Cuprins

Capitolul 1

Introducere

Capitolul 2

Preliminarii

2.1 Out-of-order Execution & Instructiuni Tranzitorii

In trecut procesoarele executau instructiunile in ordinea in care acestea erau preluate de la compilator, cate una pe rand. In multe situatii instructiuni mai costisitoare blocau fluxul de executie, iar procesorul devenea partial inactiv. Procesoarele moderne se folosesc de o serie de tehnici grupate sub umbrela *Out-of-order Execution*, introduse pentru prima data la mijlocul anilor 1990 [1], in urma unui algoritm dezvoltat de Tomasulo in 1967 [5] care permitea programarea dinamica a ordinii instructiunilor si alocarea acestora pe mai multe unitati de executie care ruleaza in paralel. Scopul acestei tehnici este utilizarea exhaustiva a resurselor disponibile pe procesor, pentru cresterea performantei.

Aceasta optimizare duce la situatii in care unele instructiuni executate trebuie respinse, iar starea programului intoarsa la una anterioara (din cauza decansarii unei exceptii in urma accesarii unei zone de memorie interzisa de exemplu). Aceste tipuri de instructiuni numite in continuare *Instructiuni Tranzitorii* stau la baza atacului *Meltdown* [3].

2.2 Branch Prediction & Executie Speculativa

Pe baza *Branch Processing Unit* (*BPU*) din interiorul procesoarelor moderne incearca sa prezica, in cazul unei ramificari (*if*), sau final de iteratie (*for, while*), ramura corecta pe care va fi urmata. In cazul in care fluxul de executie stagneaza la un astfel punct de bifurcare (de exemplu, in asteptarea incarcarii din memorie a valorii unei variabile), se poate folosi prezicerea data de *BPU* pentru a executa speculativ instructiunile urmatoare. Dupa ce executia instructiunii care decide bifurcarea este finalizata rezultatele obtinute speculativ sunt fie pastrate fie respinse [2].

Branch prediction are in general o acuratete foarte ridicata, chiar de peste 95% [1], asadar

executand speculativ s-au obtinut imbunatatiri considerabile de performanta. Cu toate acestea, in cazurile in care ramura de executie nu este prezisa corect, se vor executa instructiuni care nu ar fi avut loc in cadrul executie secventiale, *in-order execution*. Bineinteles, aceste instructiuni vor fi *rolled-back*, iar rezultatul final va fi cel asteptat, dar la nivel micro-arhitectural se pot observa si masura niste efecte neprevazute ale acestor instructiuni executate *out-of-order*. Analizarea cu grija a acestor efecte secundare sta la baza atacurilor de tip *Spectre* [2].

2.3 CPU Cache

Deoarece incarcarea valorilor din memoria RAM in cpu este foarte costisitoare, in cadrul procesoarelor exista niste zone de memorie foarte rapide, de dimensiuni reduse, ce poarta denumirea de emphcache-uri. Acestea retin valorile folosite cel mai des intr-un anumit interval de timp. Prin retinerea si citirea valorilor din cache, se mascheaza incarcarea initial relativ lenta si se castiga timp pretios de executie.

2.3.1 Atacuri asupra memoriei cache

Deoarece memoria cache este mult mai rapida, prin intermediul unui ceas de mare precizie putem distinge intre accesare din memorie si accesarea din *cache* a unei variabile. Sa consideram urmatorul exemplu:

```
uint32_t value = 10;
addr = &value;

time = __rdtscp(&junk);
junk = *addr;
// prima accesare din memorie
memory_time = __rdtscp(&junk) - time;

addr = &value;
time = __rdtscp(&junk);
junk = *addr;
// a doua accesare din cache
cache_time = __rdtscp(&junk) - time;
```

Timpul de accesare al valorii corespunzatoare variabilei **value** poate fi calculat utilizand instructiunea `__rdtscp` specifica procesoarelor Intel. Aceasta permite citirea *time-stamp counter-ului* din procesor [4]. Prin doua masuratori ce incadreaza dereferentierea pointer-ului catre `value`, putem

masura numarul de ciclii de procesor necesari operatiei. Repetand experimentul de 10000 de ori si calcularea mediei timpului de acces pentru fiecare caz, obtin urmatoarele:

- incarcarea din memorie dureaza aproximativ 250 de ciclii
- incarcarea din cache dureaza aproximativ 23 de ciclii

Diferente considerabile precum acestea sunt exploatate in cadrul diferitelor tehnici de atac asupra memoriei cache.

Capitolul 3

Continut

Capitolul 4

Concluzii

Bibliografie

[1] Joel Hruska, *What Is Speculative Execution?*, Accessed: 12.05.2022, 2021, URL: https://www.extremetech.com/computing/261792-what-is-speculative-execution.

[2] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher et al., „Spectre attacks: Exploiting speculative execution", în *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019, pp. 1–19.

[3] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom și Mike Hamburg, „Meltdown", în *arXiv preprint arXiv:1801.01207* (2018).

[4] *RDTSCP — Read Time-Stamp Counter and Processor ID*, Accesat: 12.05.2022, URL: https://www.felixcloutier.com/x86/rdtscp.

[5] Robert M Tomasulo, „An efficient algorithm for exploiting multiple arithmetic units", în *IBM Journal of research and Development* 11.1 (1967), pp. 25–33.