PQC FOR DNSSEC

The Good, the Bad and the Ugly

Ondřej Surý, ISC & Ostravská Universita; CSNOG 2026 Zlín

HROZBA JMÉNEM KVANTOVÝ POČÍTAČ

- Současné algoritmy (RSA, ECDSA, Ed25519) spoléhají na matematické problémy, které jsou pro klasické počítače těžké, ale pro kvantové snadné (Shorův algoritmus).
- S příchodem dostatečně silného kvantového počítače by se zhroutila důvěra v DNSSEC (možnost falšovat podpisy).

CO JE PQC (POST-QUANTUM CRYPTOGRAPHY)?

- Nová generace kryptografických algoritmů (na bázi mřížek, hashů, kódů, izogenií). Některé z těchto algoritmů jsou "staré".
- Navrženy tak, aby odolaly i útokům kvantových počítačů.
- Nejde o "kvantovou kryptografii" (QKD), ale o matematiku běžící na klasickém HW.

PROČ JE PQC PROBLÉM PRO DNSSEC?

- Velikost: PQC klíče a podpisy jsou často výrazně větší než ty současné.
- Limit UDP: DNS protokol má přísné limity na velikost paketu (fragmentace, 1232/1452 bytů).
- Otázka dne: Jak nacpat velké PQC podpisy do malých DNS paketů a nezničit při tom výkon internetu?

PQC PRO DNSSEC

- HAWK – využívá složitosti problémů v mřížkách (lattices), jako je hledání nejkratšího vektoru.
- SQISign – využívá zobrazení (izogenie) mezi eliptickými křivkami nad konečnými tělesy.
- MAYO – využívá obtížnosti řešení soustav polynomiálních rovnic o více proměnných.
- Antrag – Espitau, Thomas, Thi Thu Quyen Nguyen, Chao Sun, Mehdi Tibouchi, and Alexandre Wallet. 'Antrag: Annular NTRU Trapdoor Generation', 2023. https://eprint.iacr.org/2023/1335.

PQC FOR DNSSEC

ALGORITHM	NIST Level	SECRET KEY	PUBLIC KEY	SIGNATURE
FALCON-512	1	1281	897	666
HAWK-256	Challenge	96	450	249
HAWK-512	1	184	1024	555
SQIsign	1	353	65	148
MAYO	1	24	1420	454
ANTRAG-512	1	59392	768	592
RSA 2048	n/a	1232	256	256
ECDSAP256	n/a	32	64	64
ED25519	n/a	32	32	64

STAV IMPLEMENTACE

- FALCON – implementace v PQclean; embedded; chtělo to trochu poladit
- HAWK – šikovný a úsporný kód, zamýšlený přímo pro embedded 🙏
- SQISign – (zatím) nevhodné pro externí použití, CMake projekt jsem splácal tak, aby se sestavily sdílené knihovny, obecná implementace + specializovaná pro broadwell architekturu 😭
- MAYO – stejná písnička, tým se soustředí na NIST, splácáno do sdílené knihovny 😭
- ANTRAG – není to sdílená knihovna, rozhodně ještě ve fázi vývoje 😭

METODIKA LOKÁLNÍHO TESTOVÁNÍ

- System 76 Meerkat
  - Intel(R) Core(TM) Ultra 7 155H (22-vláken)
  - HT vypnuto, Turbo-Boost vypnut (takže v podstatě využívám jen **6 jader**)
- Měřeno pomocí Hyperfine
- Pro ukládání souborů použit tmpfs
- Použita «ref» implementace (nikoliv verze v assembleru)

IMPLEMENTACE V BIND 9

- RSA 2048, ECDSA255, Ed25519 – větev *ondrej/pqc-main*; velikost EDNS(0) bufferu navýšena na 1452
- BASE – větev *ondrej/pqc-base* branch; vlastní root (kořenový) server
- FALCON-512 – větev *ondrej/pqc-falcon-512*, vložená varianta z PQclean s paddingem
- HAWK-256 – větev *ondrej/pqc-hawk-256*, vložené zdrojové kódy
- HAWK-512 – větev *ondrej/pqc-hawk-512* vložené zdrojové kódy; navýšeny maximální buffery
- SQISign – větev *ondrej/pqc-sqisign*, vložené sdílené knihovny (jo, já vím, hrůza)
- MAYO – větev *ondrej/pqc-mayo*, vložené sdílené knihovny (fuj!); navýšeny maximální buffery
- Antrag-512 – větev *ondrej/pqc-antrag*, upravená testovací sada pro integraci a opraveny chyby v C kódu
- ML-DSA (bonus) – větev *ondrej/ml-dsa-for-dnssec*, implementace pomocí OpenSSL (novinka z ledna 2026)

GENEROVÁNÍ KLÍČŮ

ALGORITHM	MEAN	σ
FALCON-512	80.1 ms	11.7 ms
HAWK-256	46.9 ms	1.2 ms
HAWK-512	*51.5 ms*	3.5 ms
SQISign	97.8 ms	4.4 ms
MAYO	45.1 ms	2.9 ms
ANTRAG-512	71.9 ms	2.6 ms
RSA 2048	493.7 ms	253.8 ms
ECDSAP256	45.1 ms	2.3 ms
ED25519	45.2 ms	2.3 ms

PODPISY (ROOT, 1 KSK, 1 ZSK, RAW)

ALGORITHM	MEAN	σ	SIGNATURES/S	RAW SIZE
FALCON-512	4881.9 ms	26.8 ms	589	2891700
HAWK-256	195.5 ms	4.9 ms	62001	1727793
HAWK-512	261.0 ms	9.6 ms	49821	2582375
SQISign	*54528.1 ms*	67.9 ms	51	1445334
MAYO	1086.6 ms	48.7 ms	2746	2301478
ANTRAG-512[+]	5339.6 ms	111.2 ms	546	2685056
RSA 2048	845.7 ms	3.0 ms	3980	1746936
ECDSAP256	218.1 ms	10.2 ms	44286	1211056
ED25519	240.6 ms	6.3 ms	47288	1210992

[+] Single threaded

VELIKOST ZPRÁV (ROOT, 1 KSK, 1 ZSK)

ALGORITHM	SOA	DNSKEY	NXDOMAIN	NODATA	Delegation
FALCON-512	797	3244	1520	1518	1023
HAWK-256	380	1237	686	684	606
HAWK-512	686	2691	1298	1296	912
SQISign	279	366	484	482	505
MAYO	1108	3382	1096	1094	811
ANTRAG-512	723	2216	1372	1370	949
RSA 2048	387	864	700	698	613
ECDSAP256	195	280	316	314	421
ED25519	195	216	316	314	421

Nevejde se do 1232!

Nevejde se do 1452!

VALIDACE (ROOT, 1 KSK, 1 ZSK, RAW)

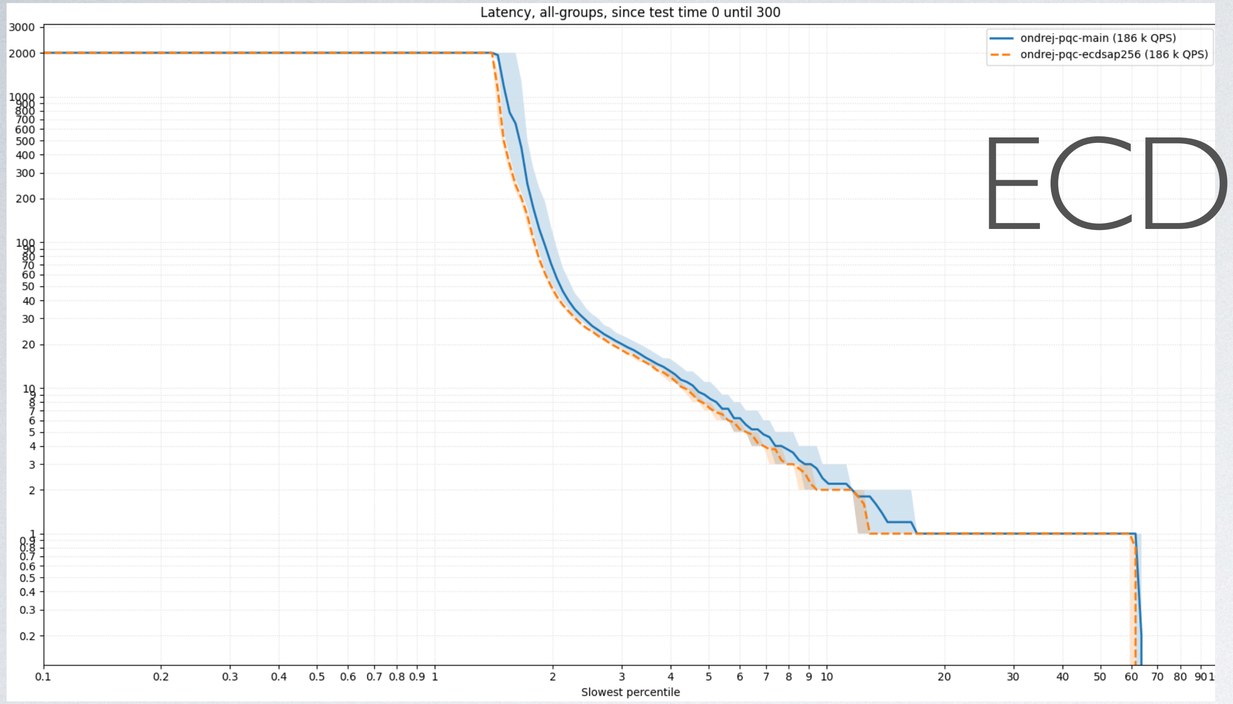
ALGORITHM	MEAN	σ
FALCON-512	403.7 ms	1.1 ms
HAWK-256	232.5 ms	1.4 ms
HAWK-512	359.4 ms	66.0 ms
SQISign	22338.5 ms	35.0 ms
MAYO	995.8 ms	26.8 ms
ANTRAG-512	548.6 ms	1.4 ms
RSA 2048	250.2 ms	18.9 ms
ECDSAP256	610.0 ms	4.5 ms
ED25519	819.4 ms	4.5 ms

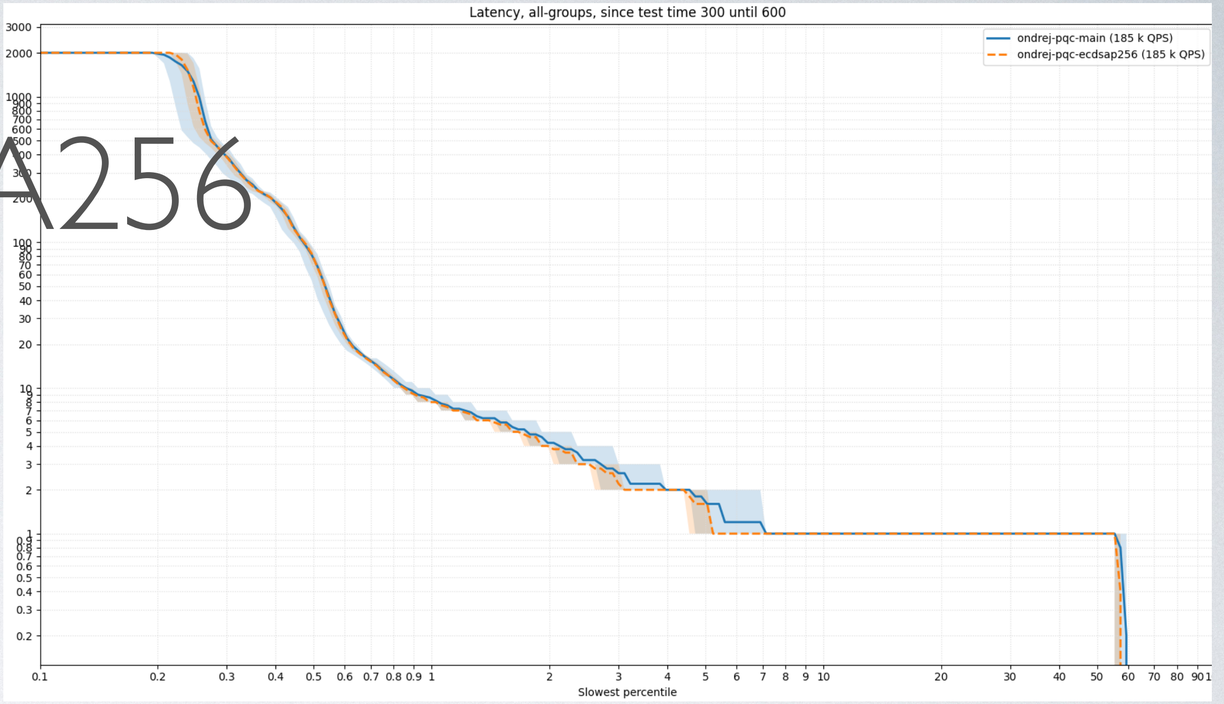
** Warning: Statistical outliers were detected.*

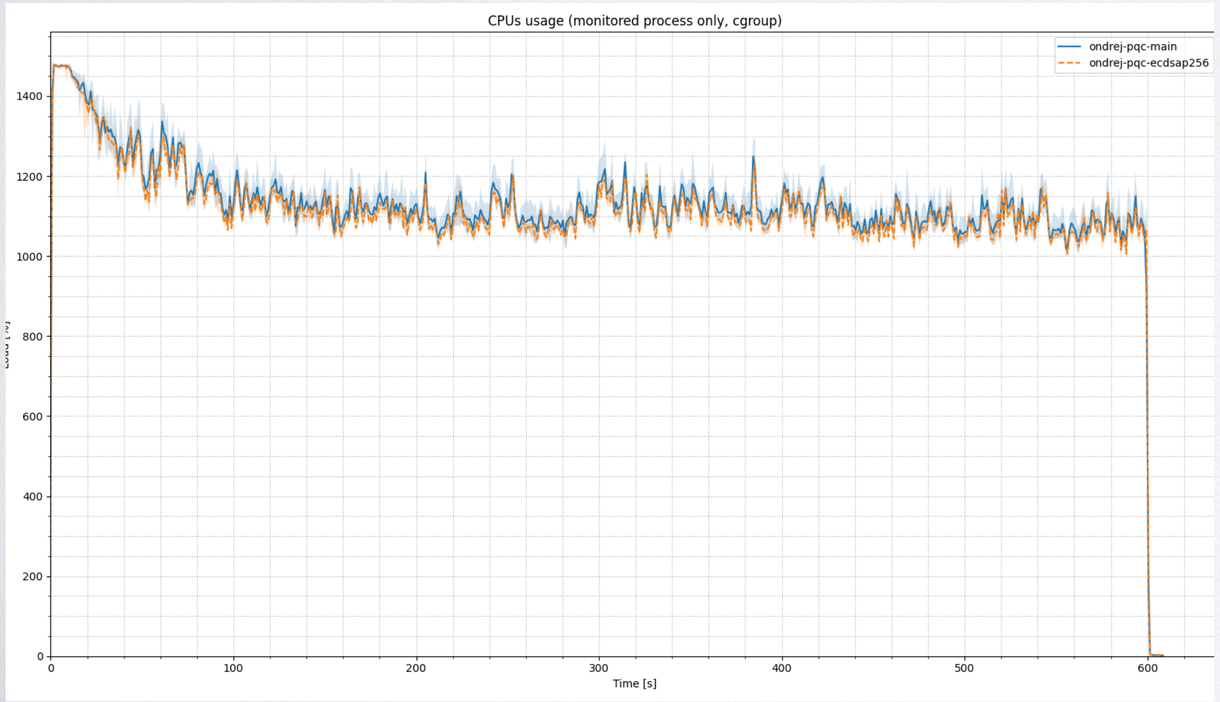
TESTOVÁNÍ RESOLVERU

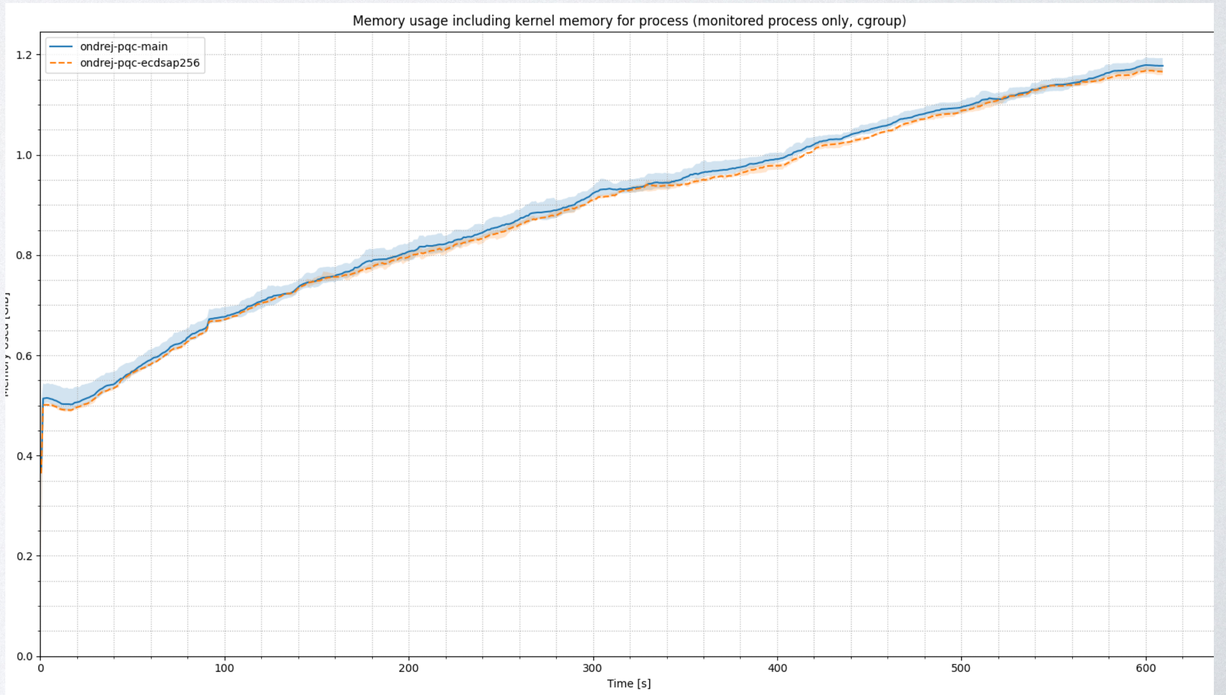
- Data z reálného provozu Resolveru (od jednoho nordického ISP)
- Vlastní kořenový (root) server obsluhující podepsanou kořenovou zónu
- Použit DNS Shotgun – součást standardního testování BIND 9

ECDSA256

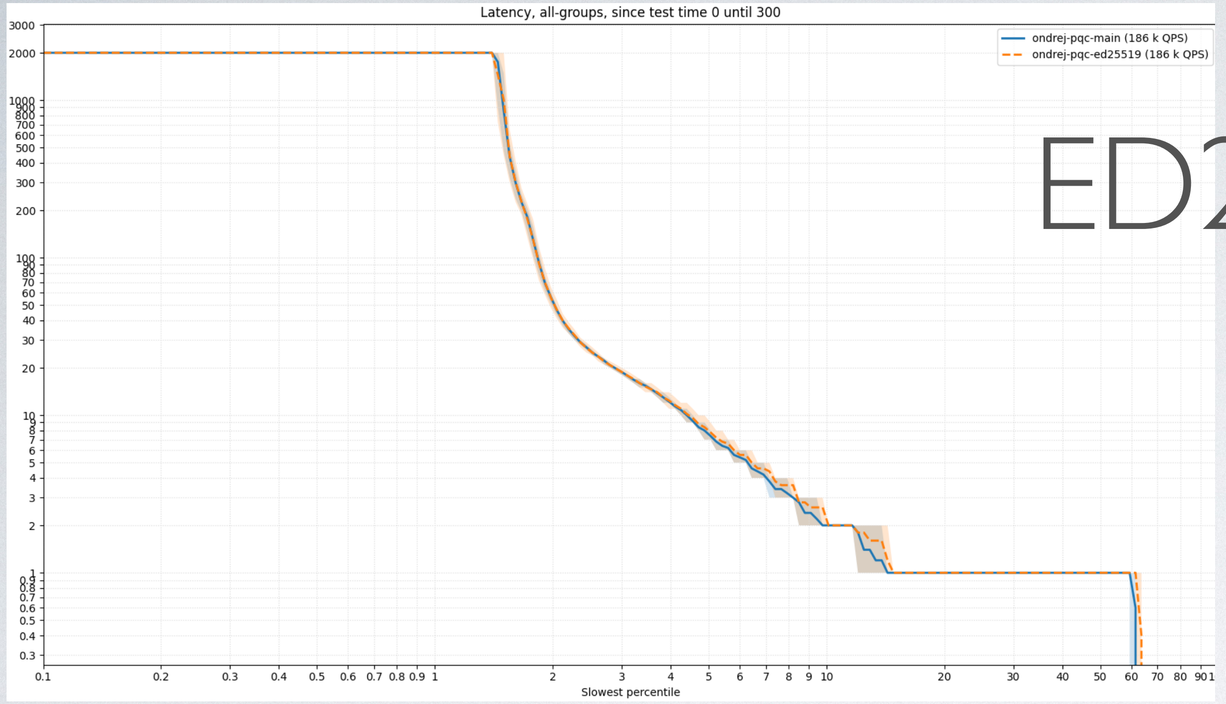

Latency, all-groups, since test time 0 until 300
ondrej-pqc-main (186 k QPS)
ondrej-pqc-ecdsap256 (186 k QPS)
Slowest percentile

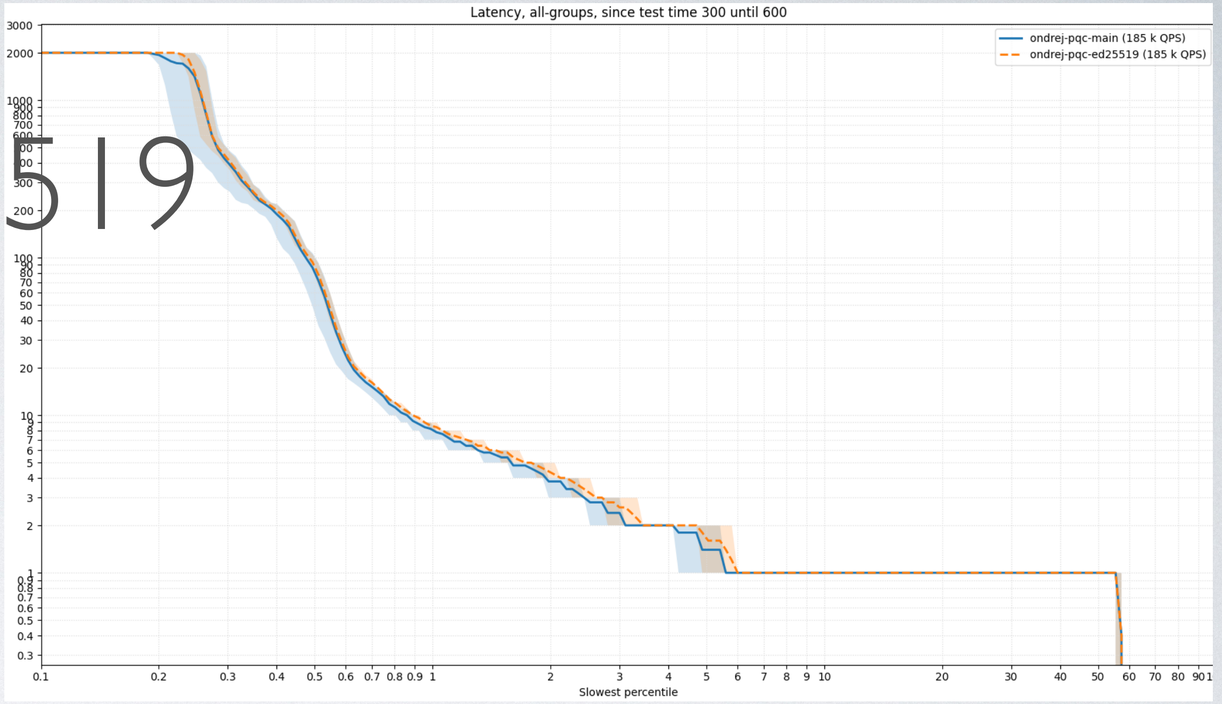

Latency, all-groups, since test time 300 until 600
ondrej-pqc-main (185 k QPS)
ondrej-pqc-ecdsap256 (185 k QPS)
Slowest percentile

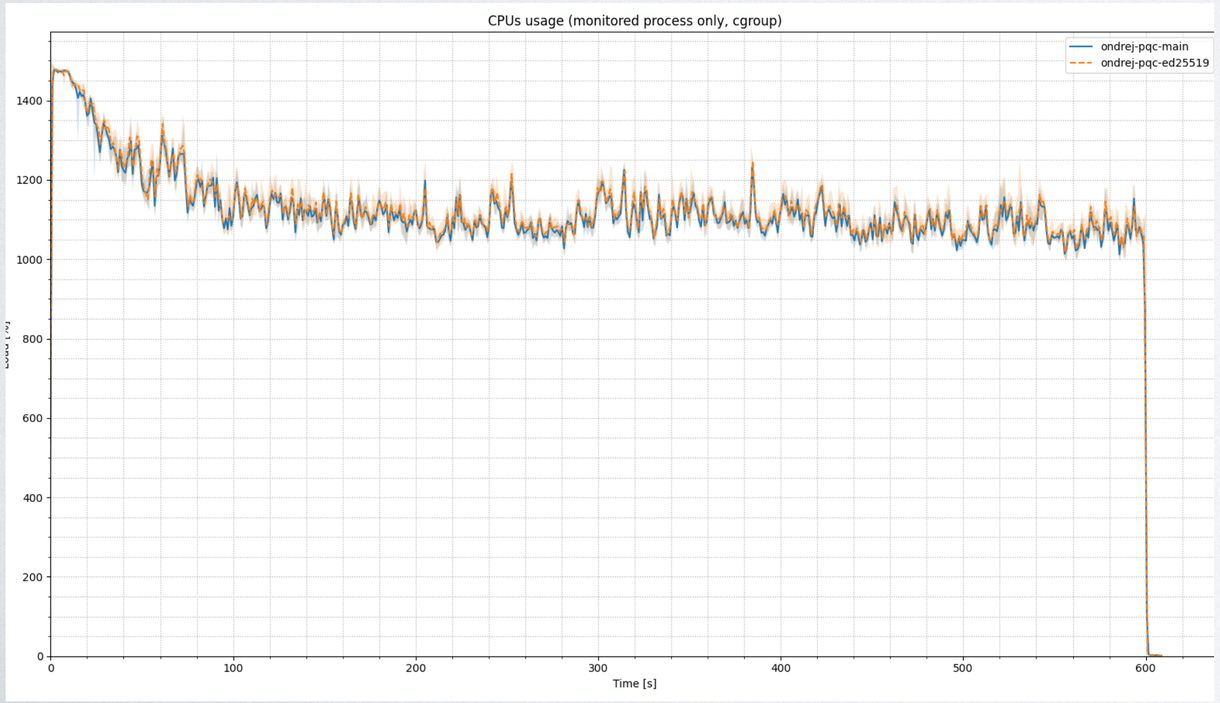

CPUs usage (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-ecdsap256
Time [s]

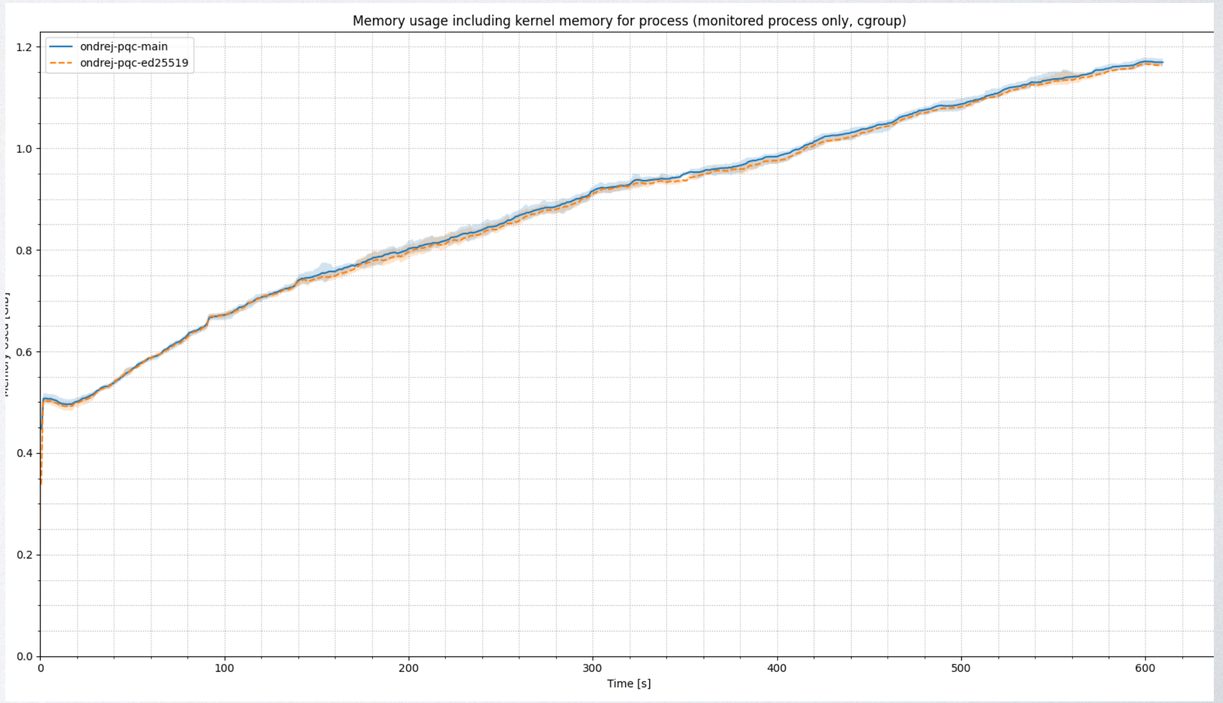

Memory usage including kernel memory for process (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-ecdsap256
Time [s]

ED25519

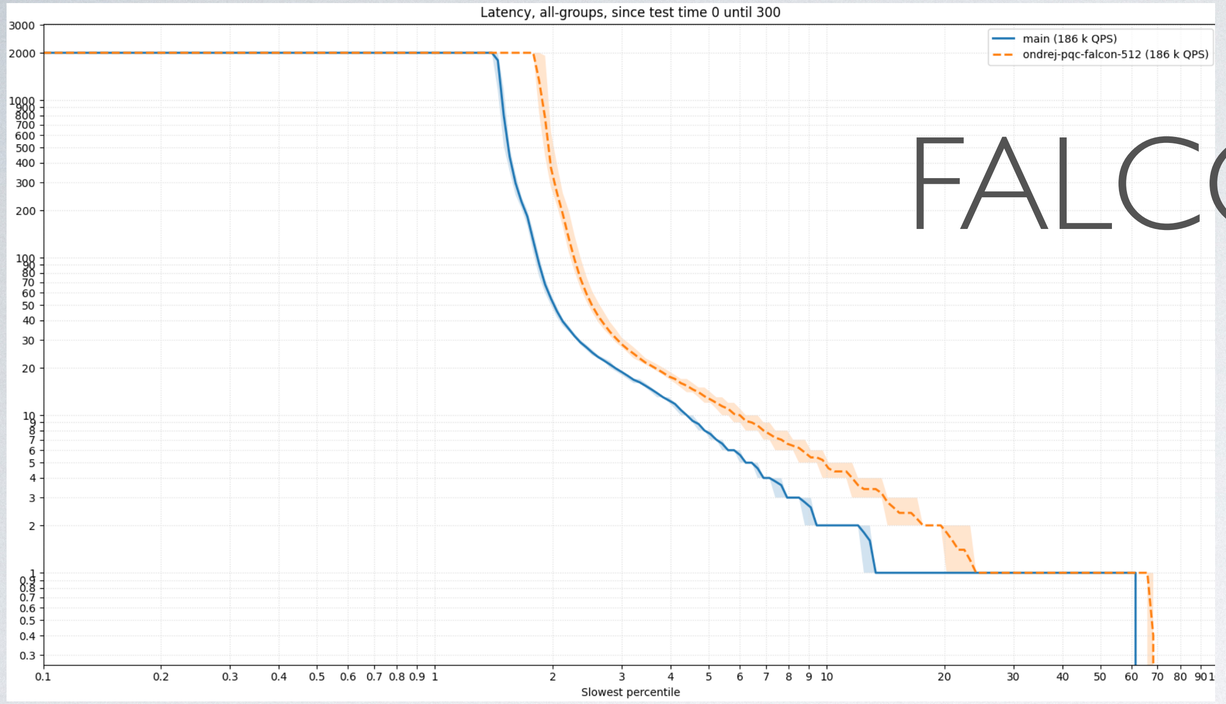

Latency, all-groups, since test time 0 until 300
ondrej-pqc-main (186 k QPS)
ondrej-pqc-ed25519 (186 k QPS)
Slowest percentile

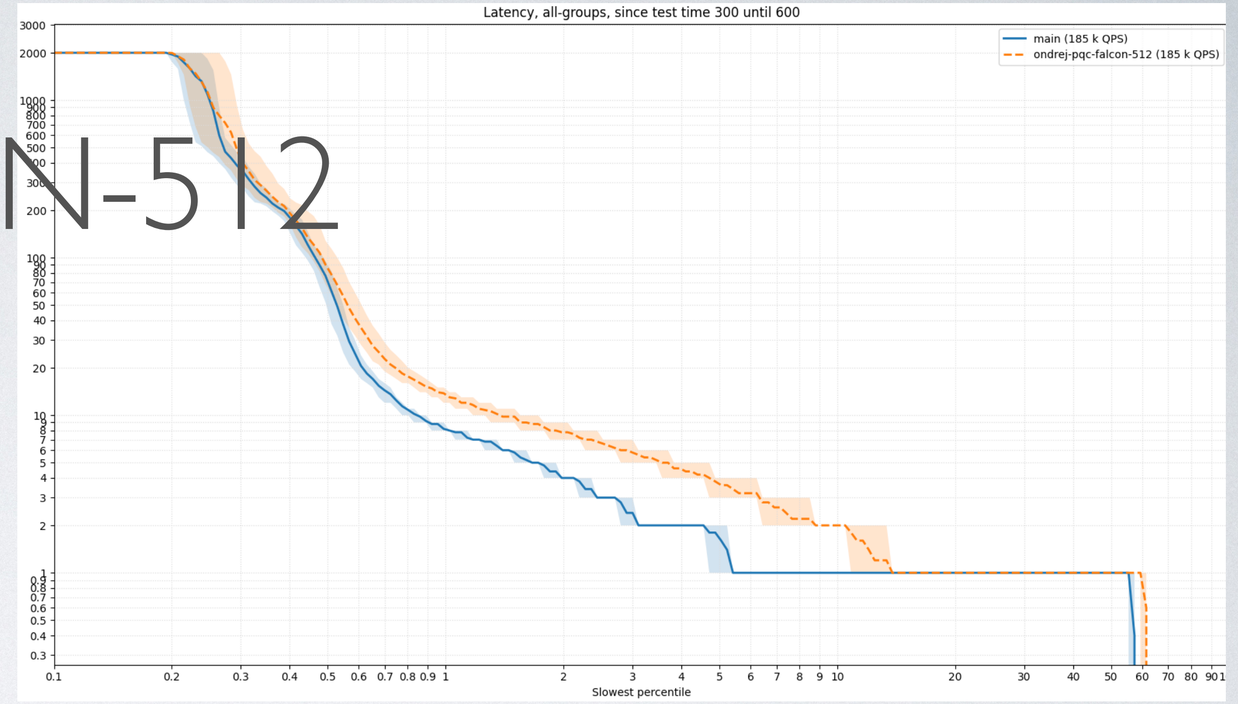

Latency, all-groups, since test time 300 until 600
ondrej-pqc-main (185 k QPS)
ondrej-pqc-ed25519 (185 k QPS)
Slowest percentile


CPUs usage (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-ed25519
Time [s]

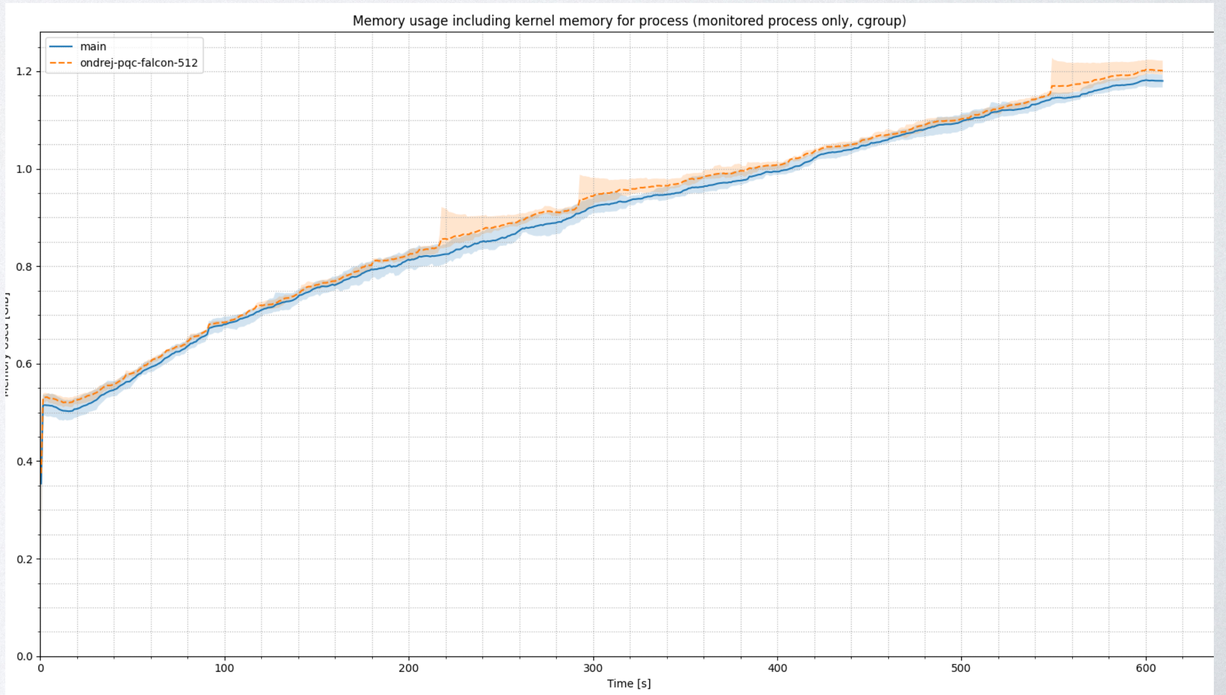

Memory usage including kernel memory for process (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-ed25519
Time [s]

FALCON-512

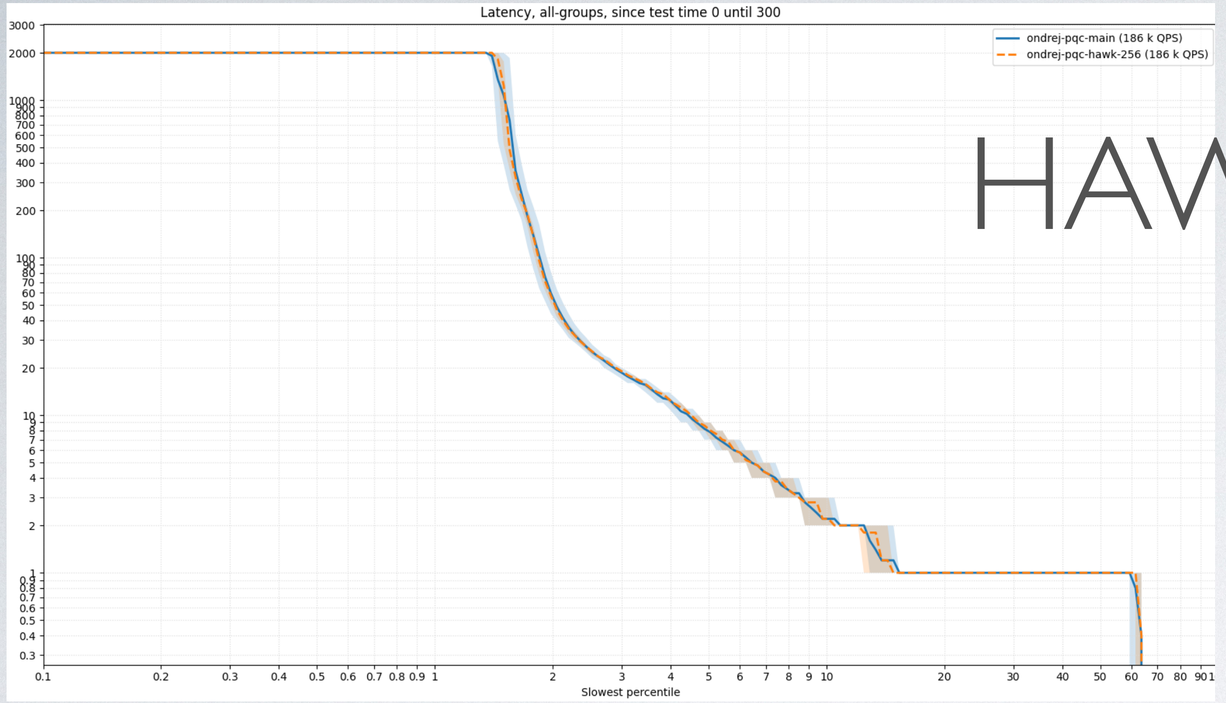

Latency, all-groups, since test time 0 until 300
main (186 k QPS)
ondrej-pqc-falcon-512 (186 k QPS)
Slowest percentile

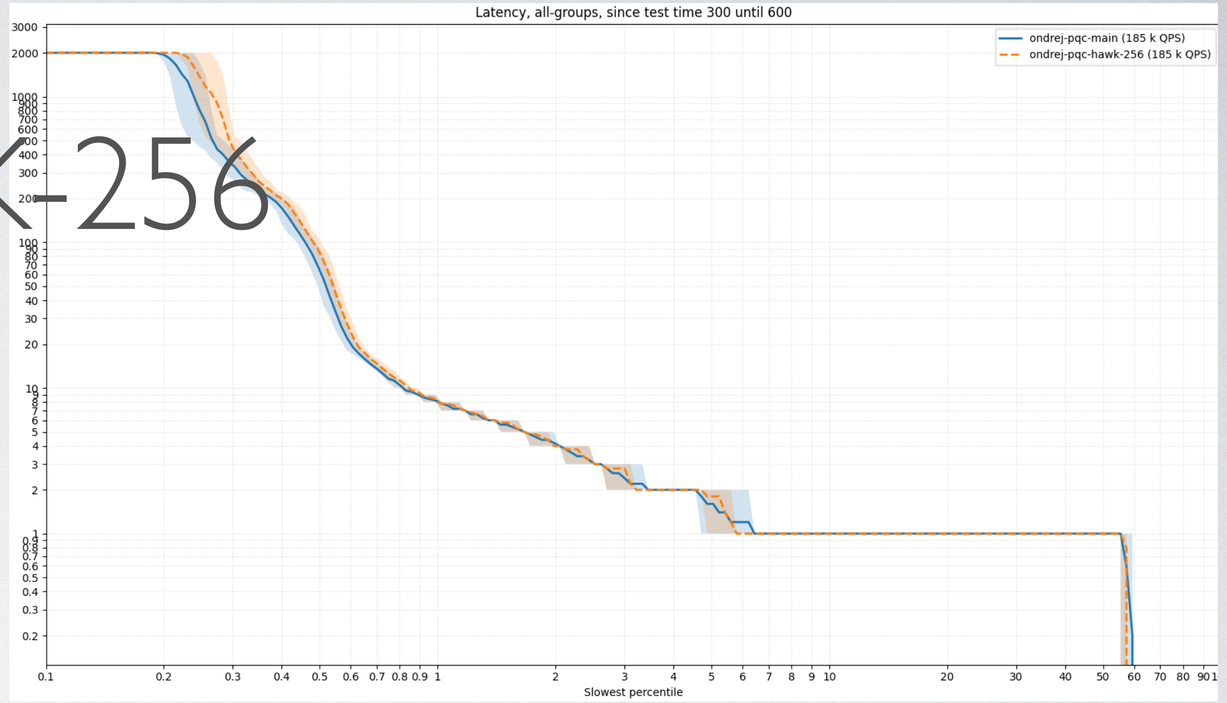

Latency, all-groups, since test time 300 until 600
main (185 k QPS)
ondrej-pqc-falcon-512 (185 k QPS)
Slowest percentile

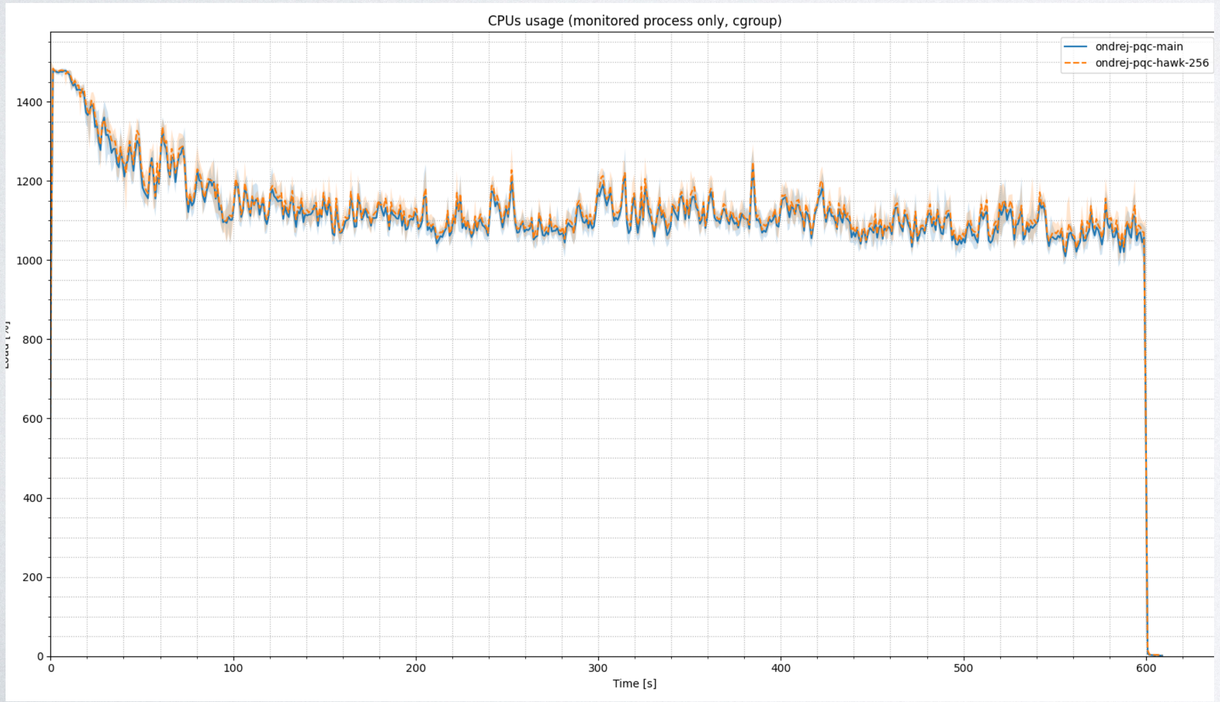

CPUs usage (monitored process only, cgroup)
main
ondrej-pqc-falcon-512
Time [s]

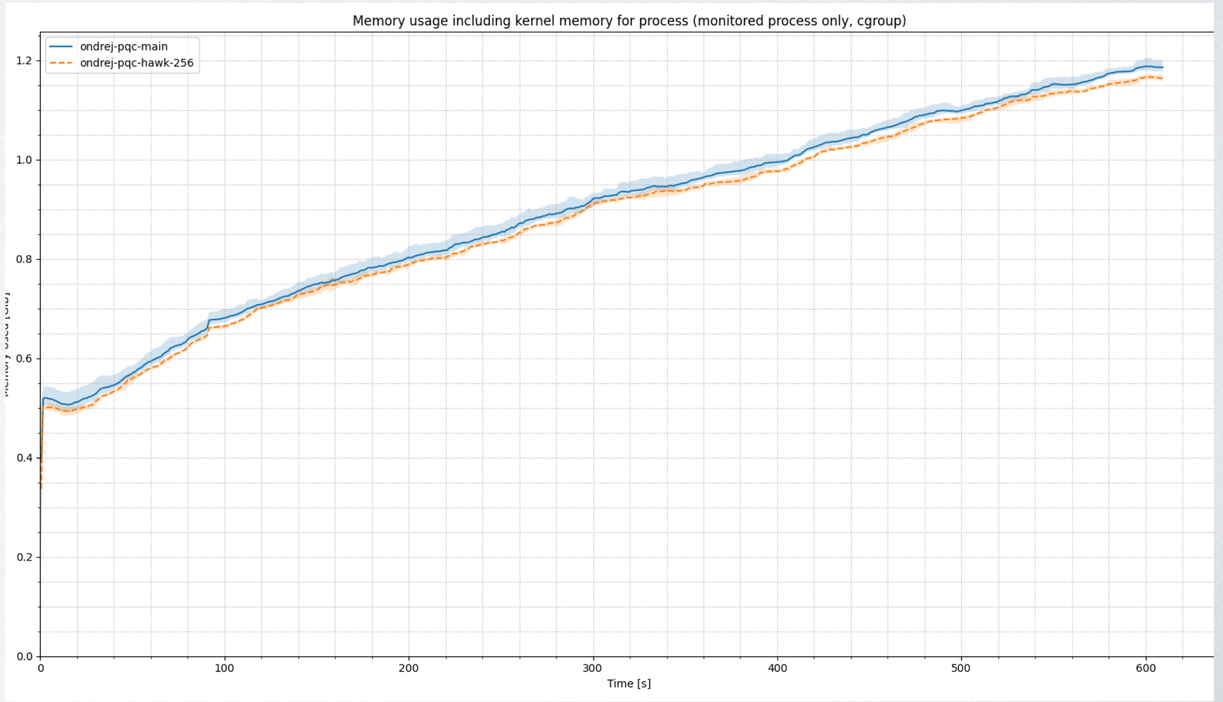

Memory usage including kernel memory for process (monitored process only, cgroup)
main
ondrej-pqc-falcon-512
Time [s]

HAWK-256

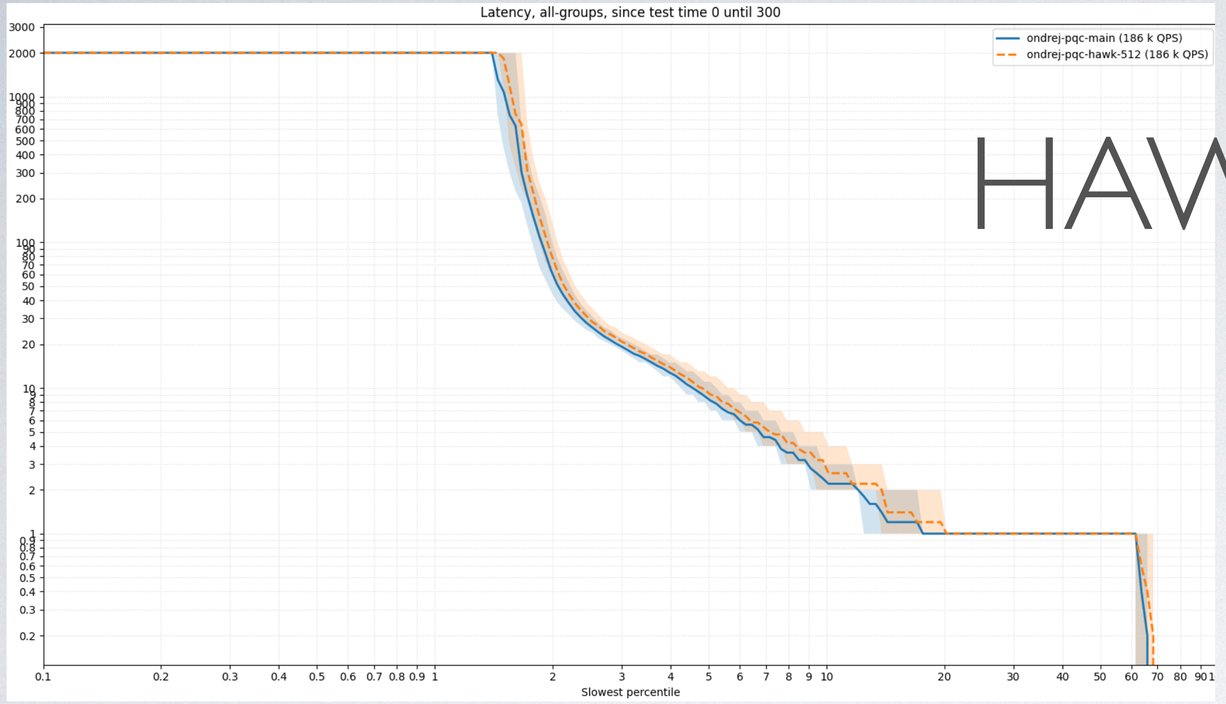

Latency, all-groups, since test time 0 until 300
ondrej-pqc-main (186 k QPS)
ondrej-pqc-hawk-256 (186 k QPS)
Slowest percentile

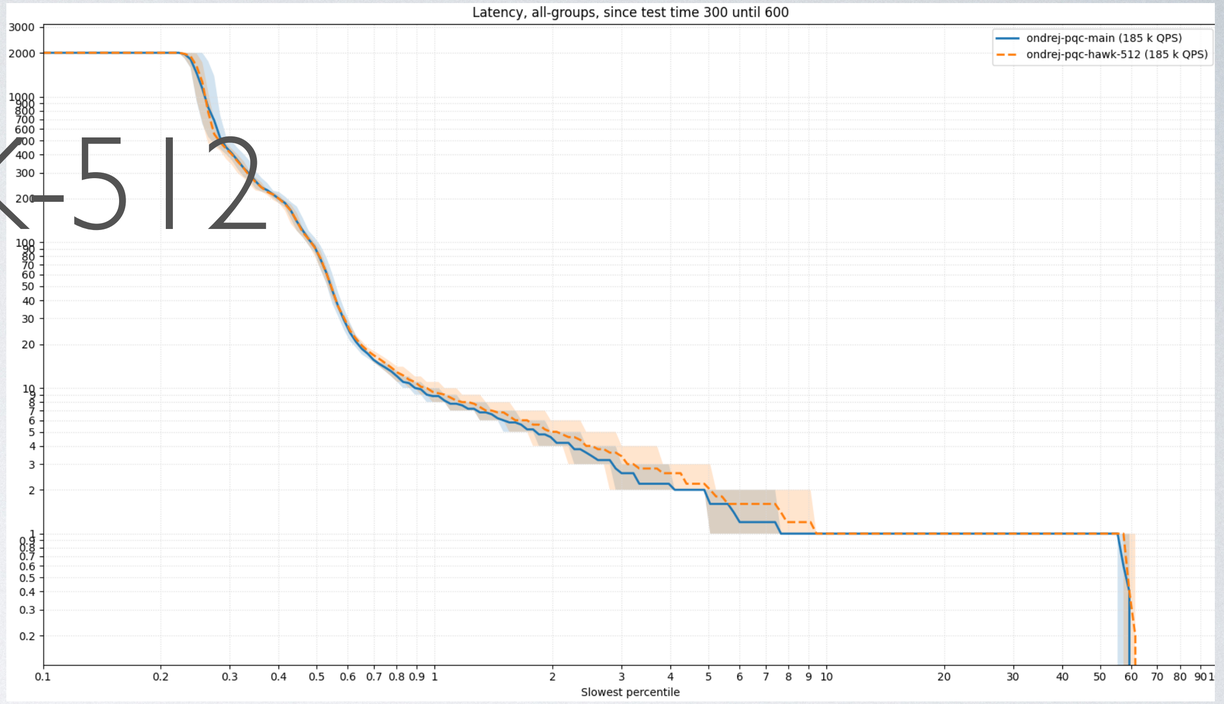

Latency, all-groups, since test time 300 until 600
ondrej-pqc-main (185 k QPS)
ondrej-pqc-hawk-256 (185 k QPS)
Slowest percentile

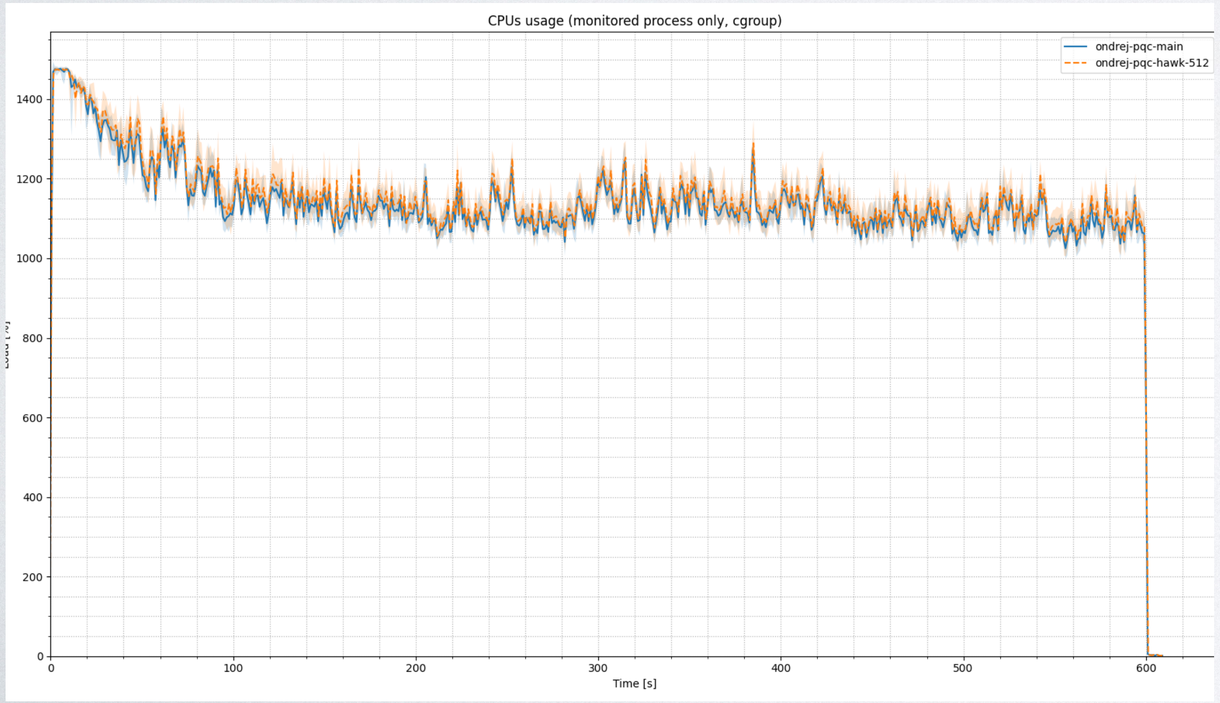

CPUs usage (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-hawk-256
Time [s]

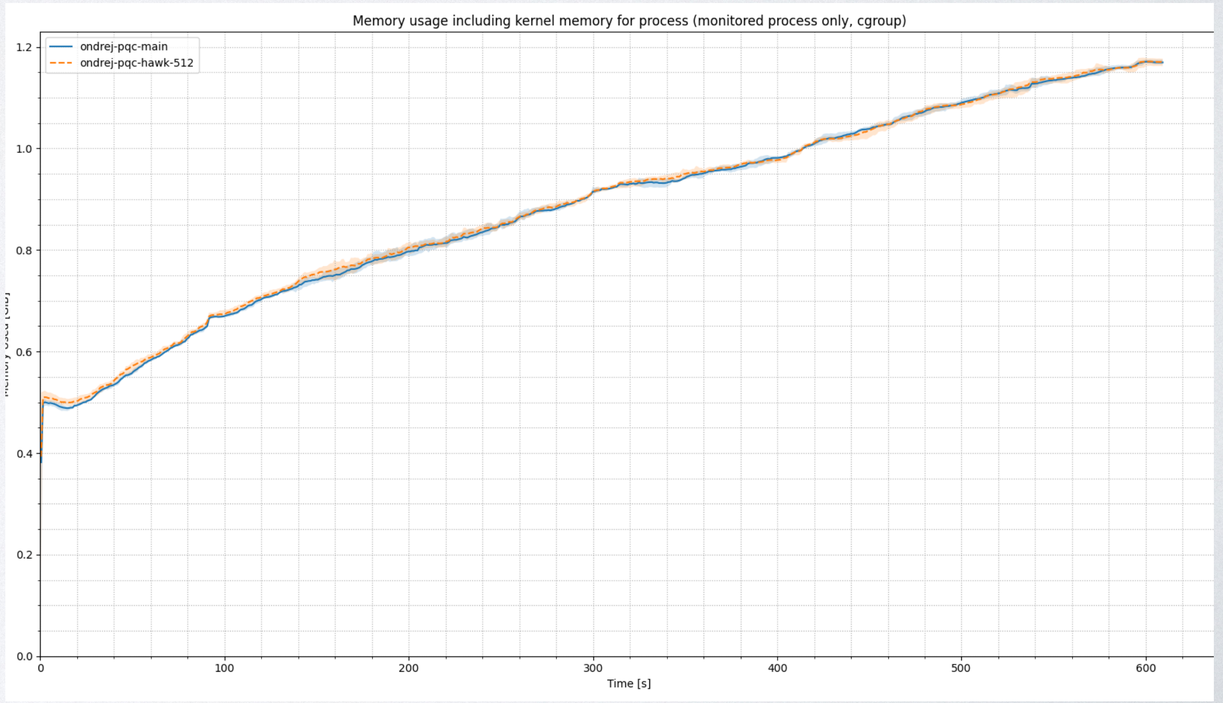

Memory usage including kernel memory for process (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-hawk-256
Time [s]

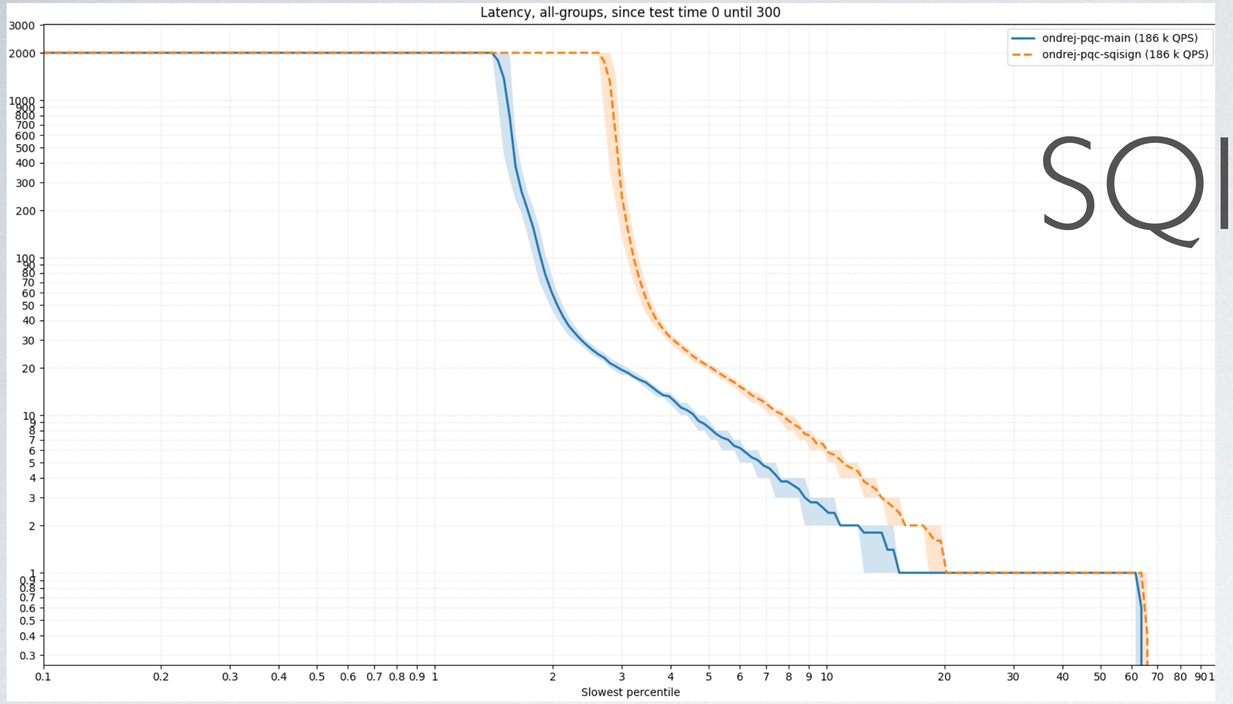
HAWK-512


Latency, all-groups, since test time 0 until 300
ondrej-pqc-main (186 k QPS)
ondrej-pqc-hawk-512 (186 k QPS)
Slowest percentile

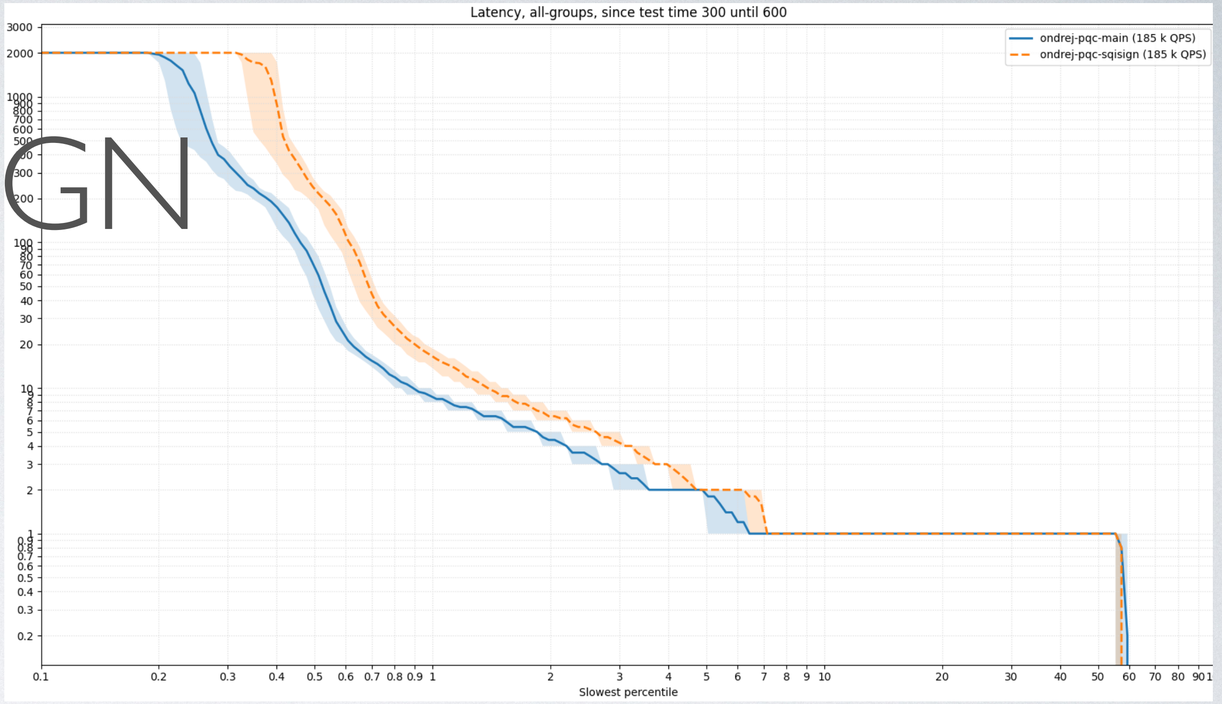

Latency, all-groups, since test time 300 until 600
ondrej-pqc-main (185 k QPS)
ondrej-pqc-hawk-512 (185 k QPS)
Slowest percentile

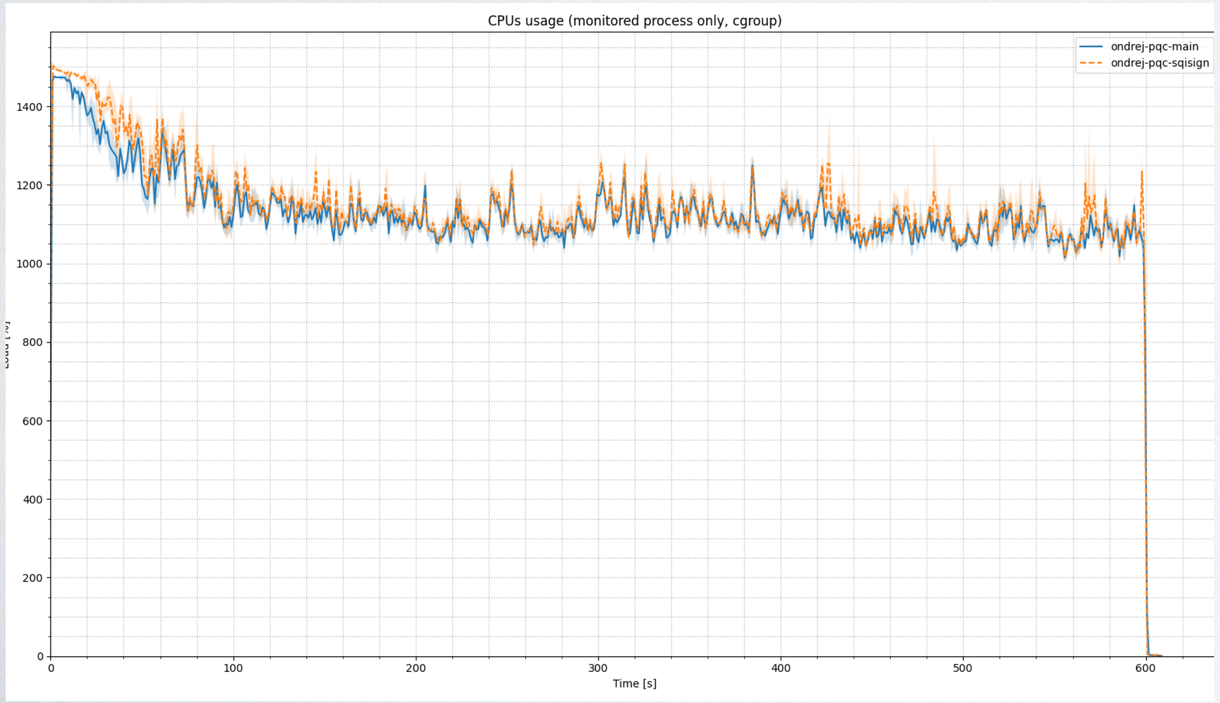

CPUs usage (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-hawk-512
Time [s]

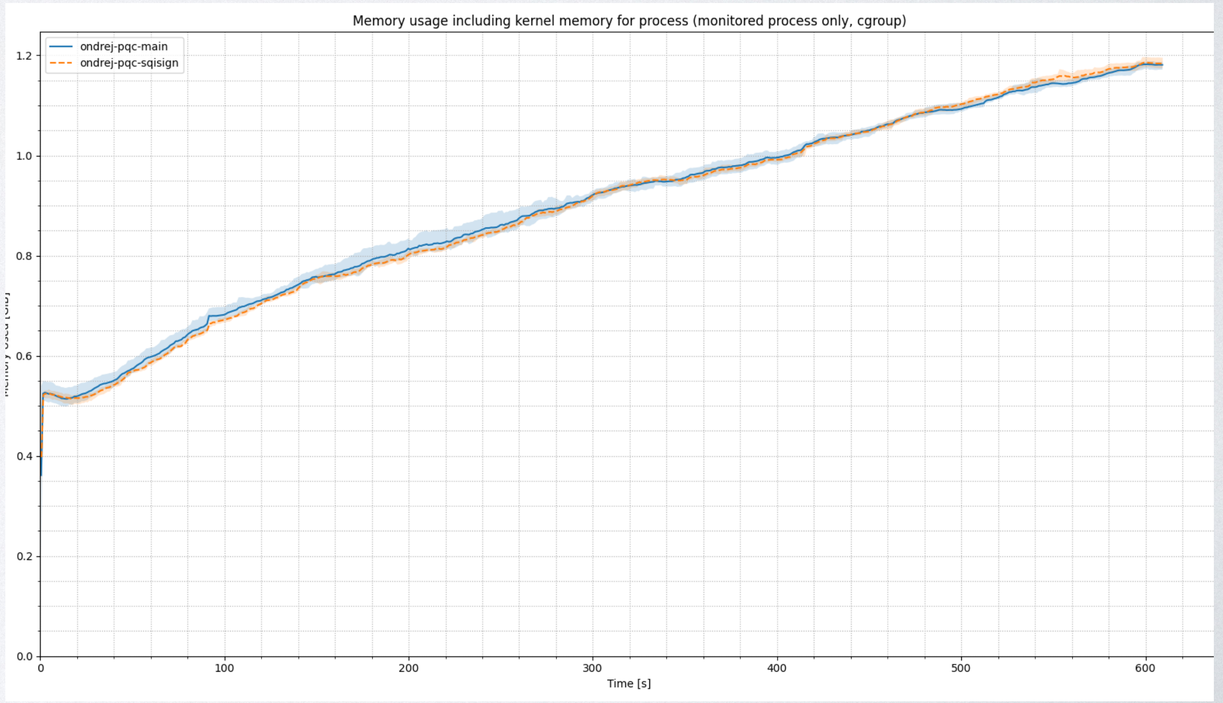

Memory usage including kernel memory for process (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-hawk-512
Time [s]

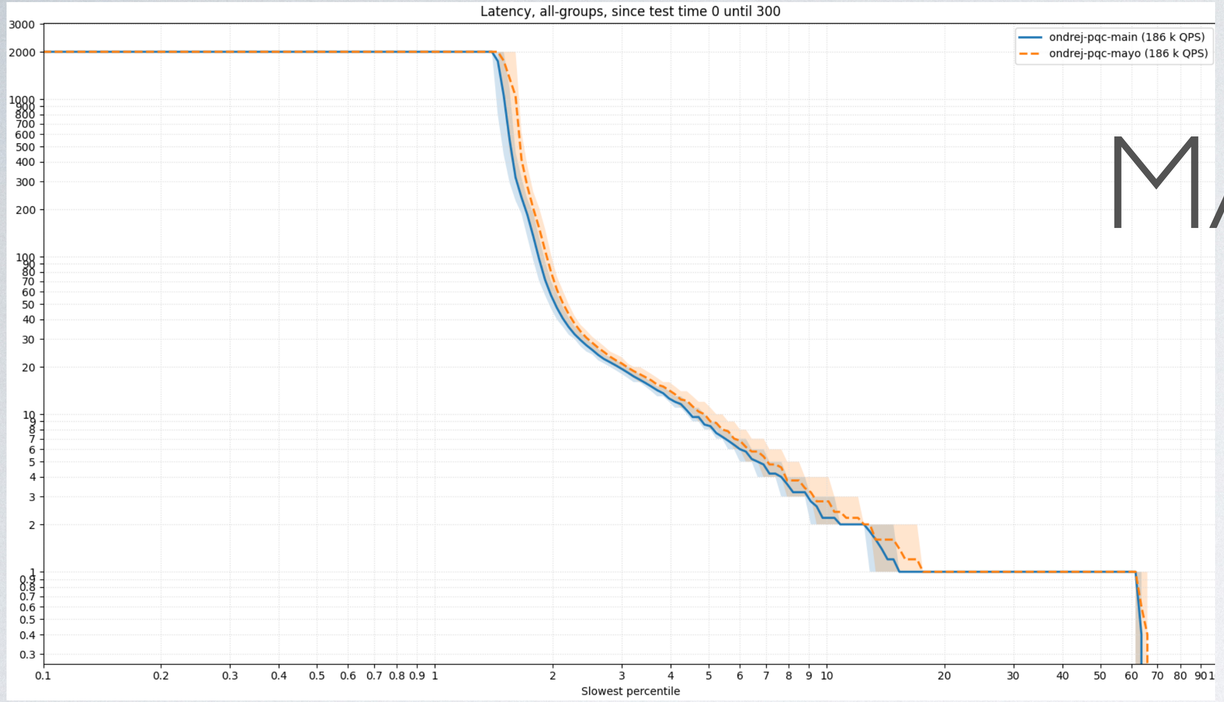

Latency, all-groups, since test time 0 until 300
ondrej-pqc-main (186 k QPS)
ondrej-pqc-sqisign (186 k QPS)
Slowest percentile

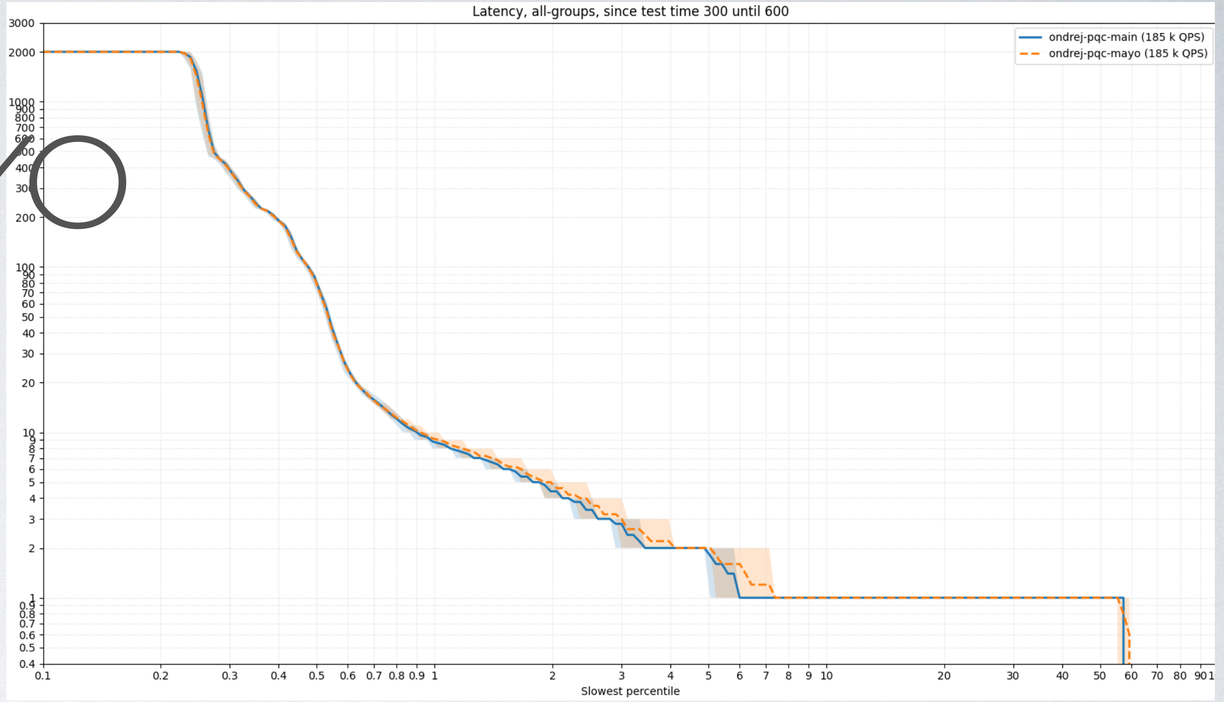
SQISIGN

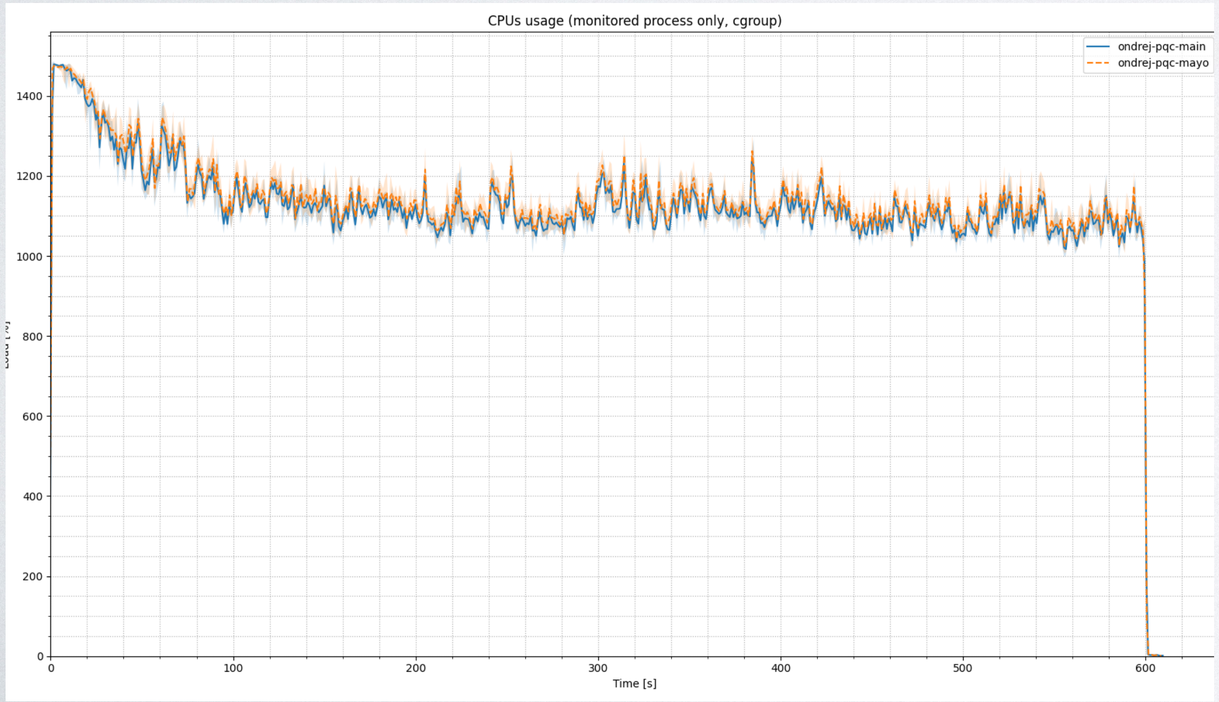

Latency, all-groups, since test time 300 until 600
ondrej-pqc-main (185 k QPS)
ondrej-pqc-sqisign (185 k QPS)
Slowest percentile

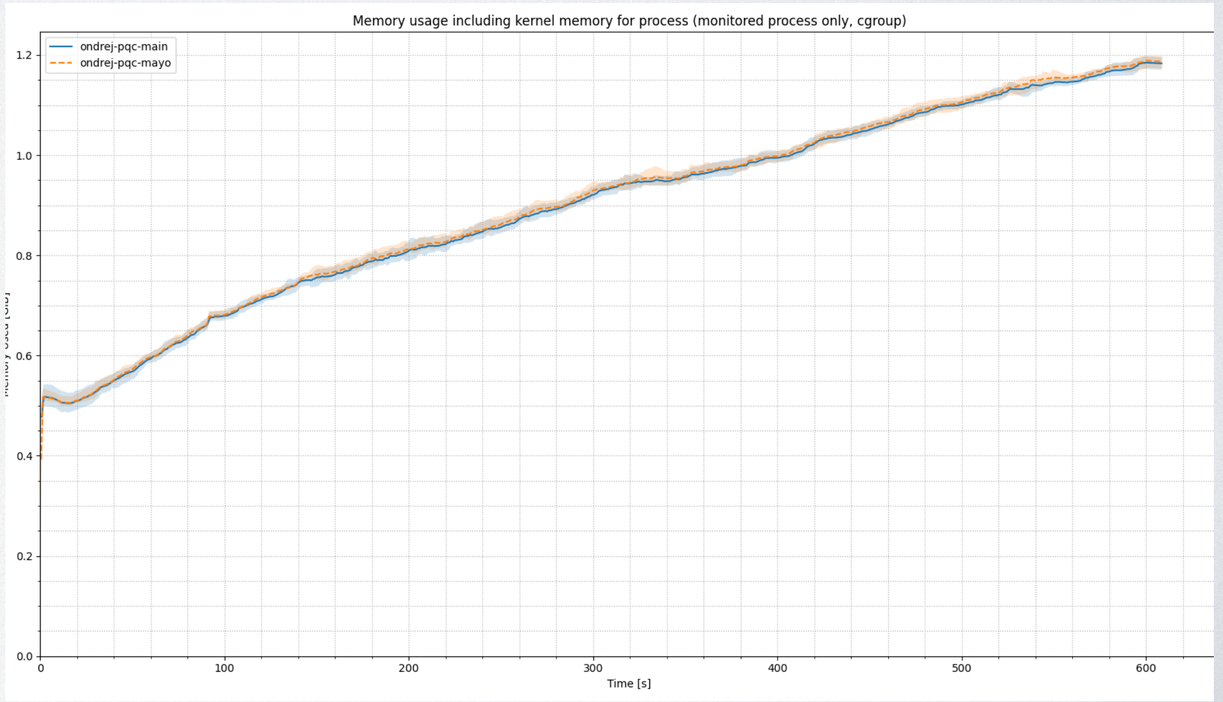

CPUs usage (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-sqisign
Time [s]


Memory usage including kernel memory for process (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-sqisign
Time [s]

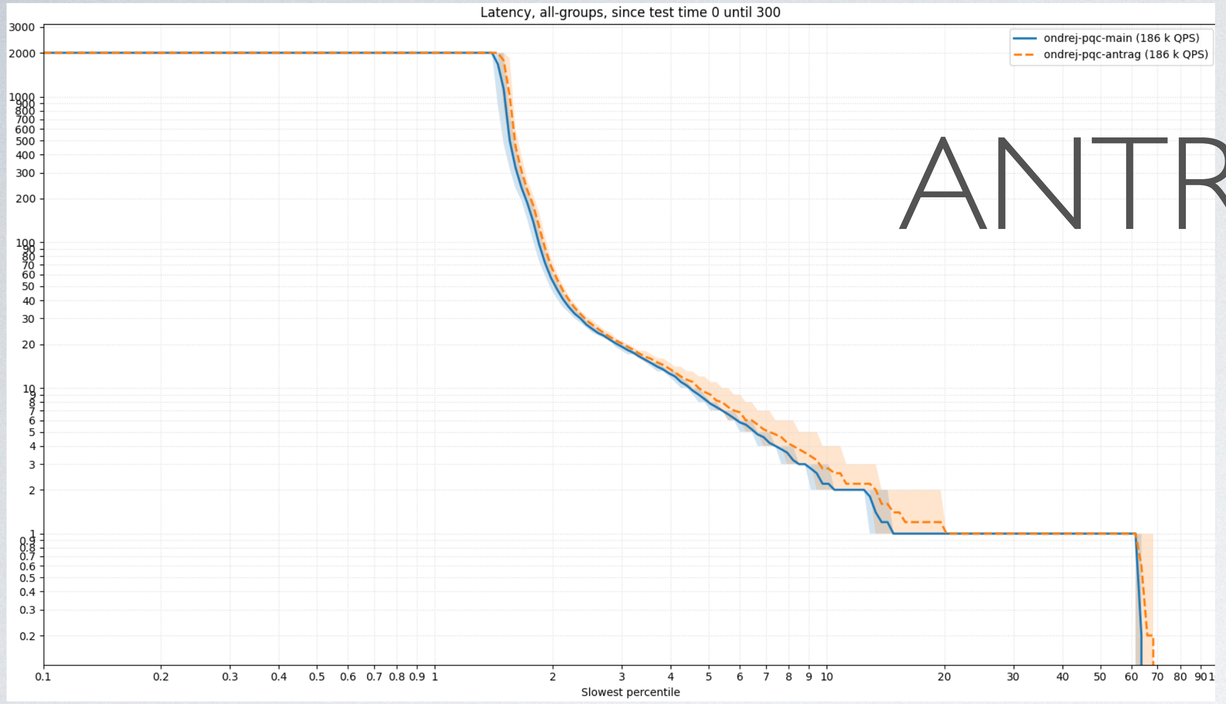

Latency, all-groups, since test time 0 until 300
ondrej-pqc-main (186 k QPS)
ondrej-pqc-mayo (186 k QPS)
Slowest percentile

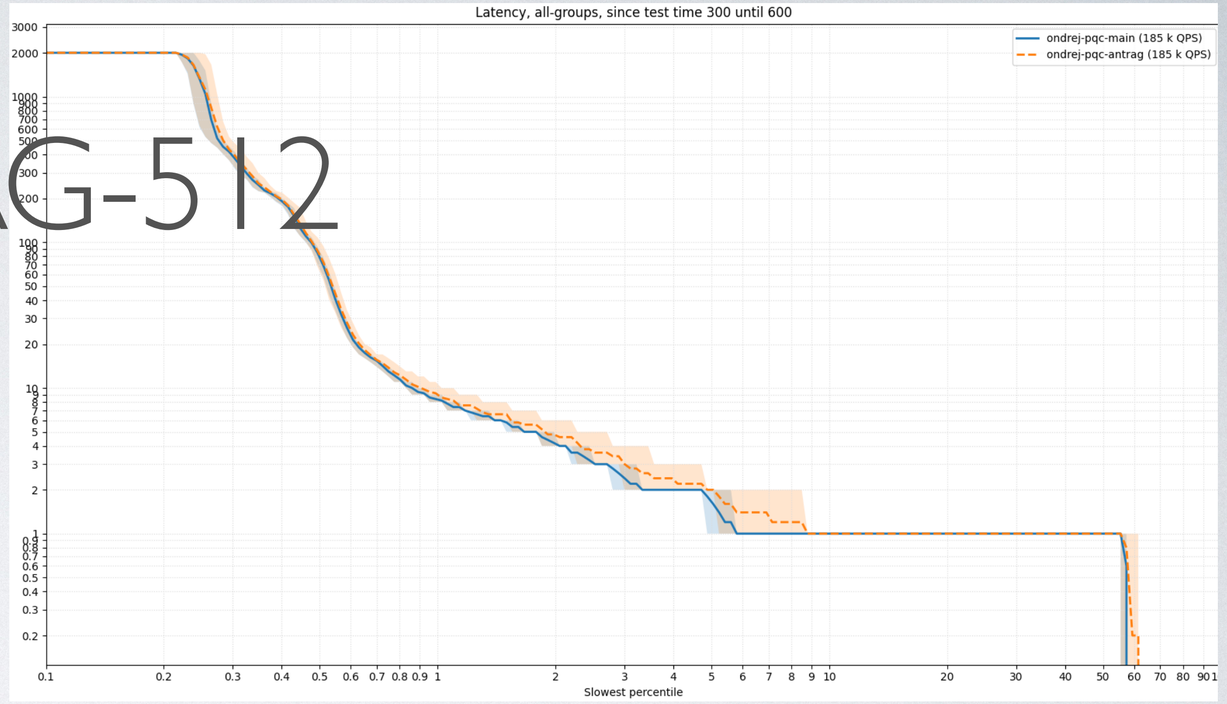

Latency, all-groups, since test time 300 until 600
ondrej-pqc-main (185 k QPS)
ondrej-pqc-mayo (185 k QPS)
Slowest percentile

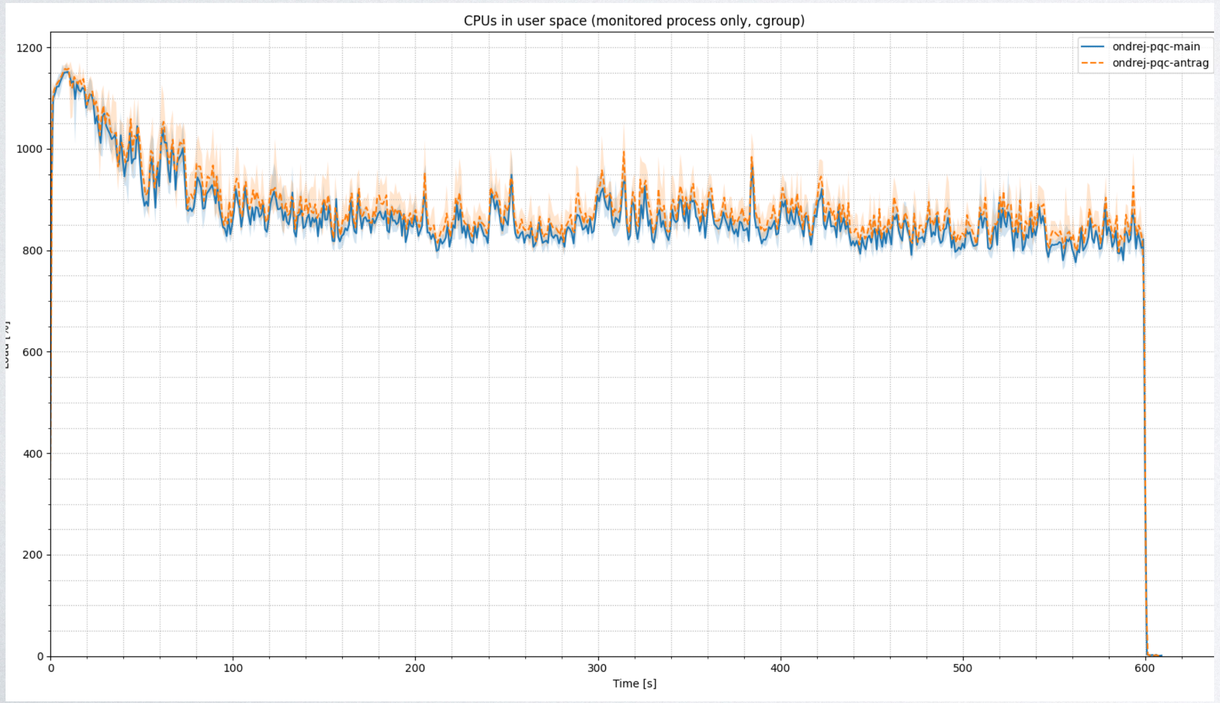

CPUs usage (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-mayo
Time [s]

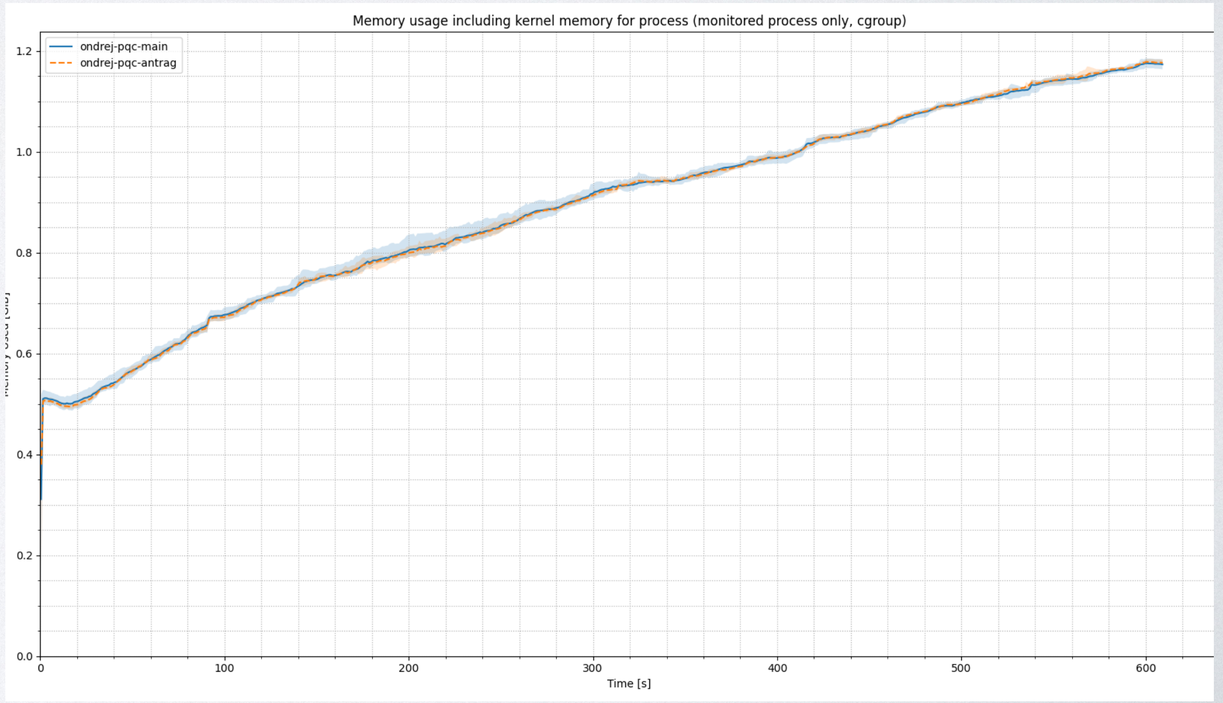

Memory usage including kernel memory for process (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-mayo
Time [s]

ANTRAG-512

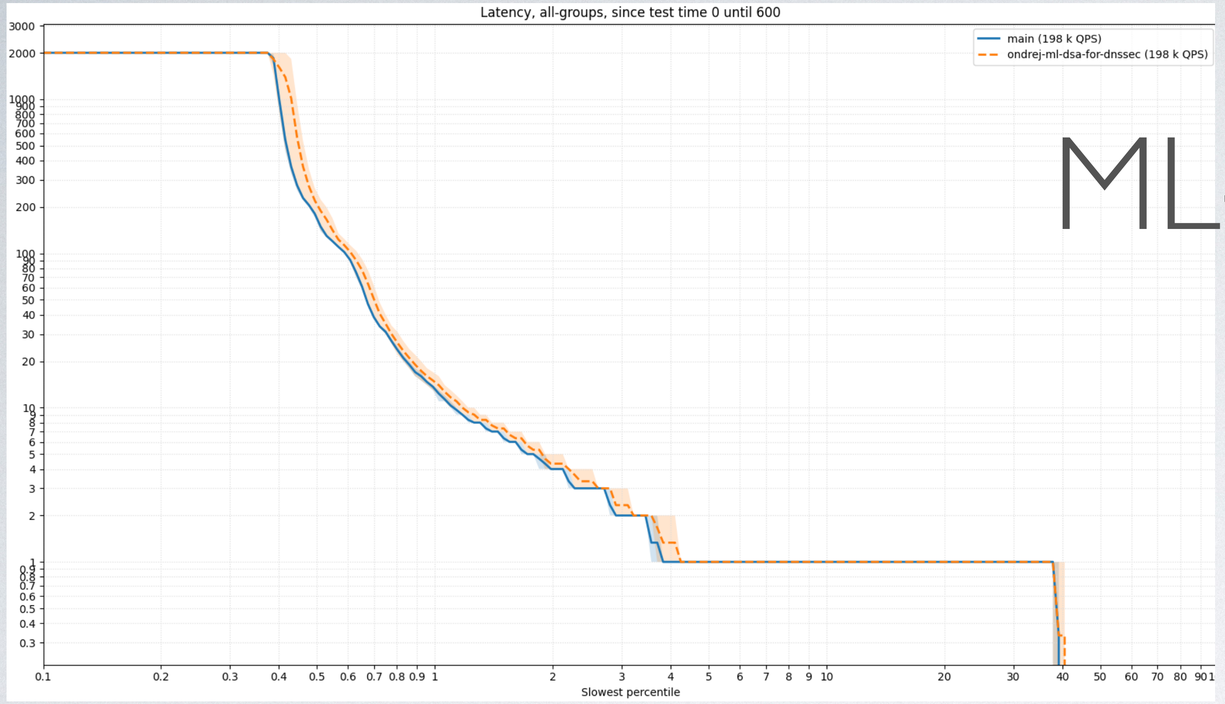

Latency, all-groups, since test time 0 until 300
ondrej-pqc-main (186 k QPS)
ondrej-pqc-antrag (186 k QPS)
Slowest percentile

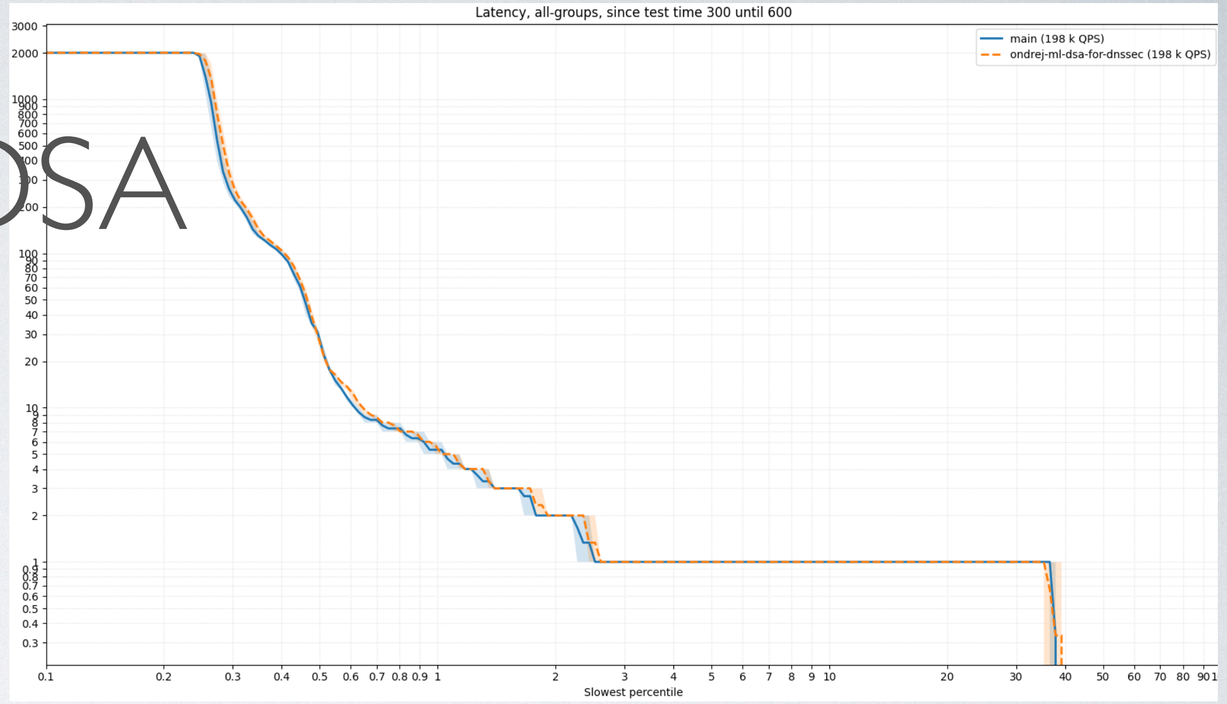

Latency, all-groups, since test time 300 until 600
ondrej-pqc-main (185 k QPS)
ondrej-pqc-antrag (185 k QPS)
Slowest percentile

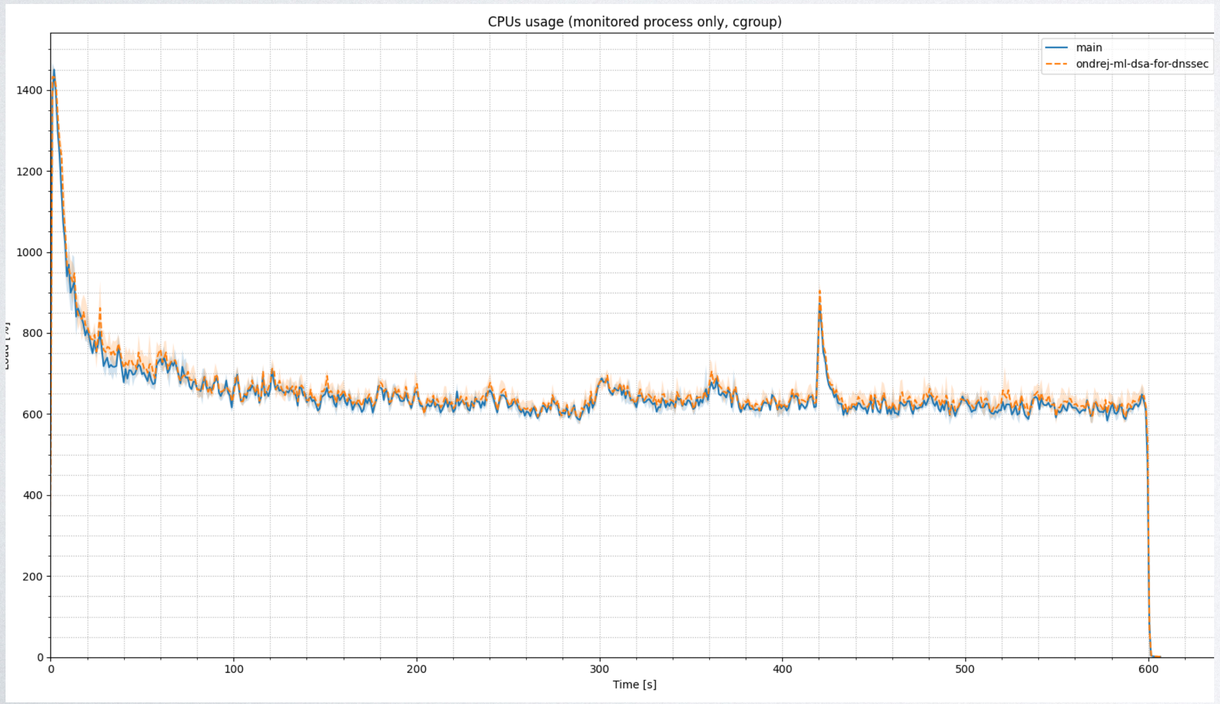

CPUs in user space (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-antrag
Time [s]

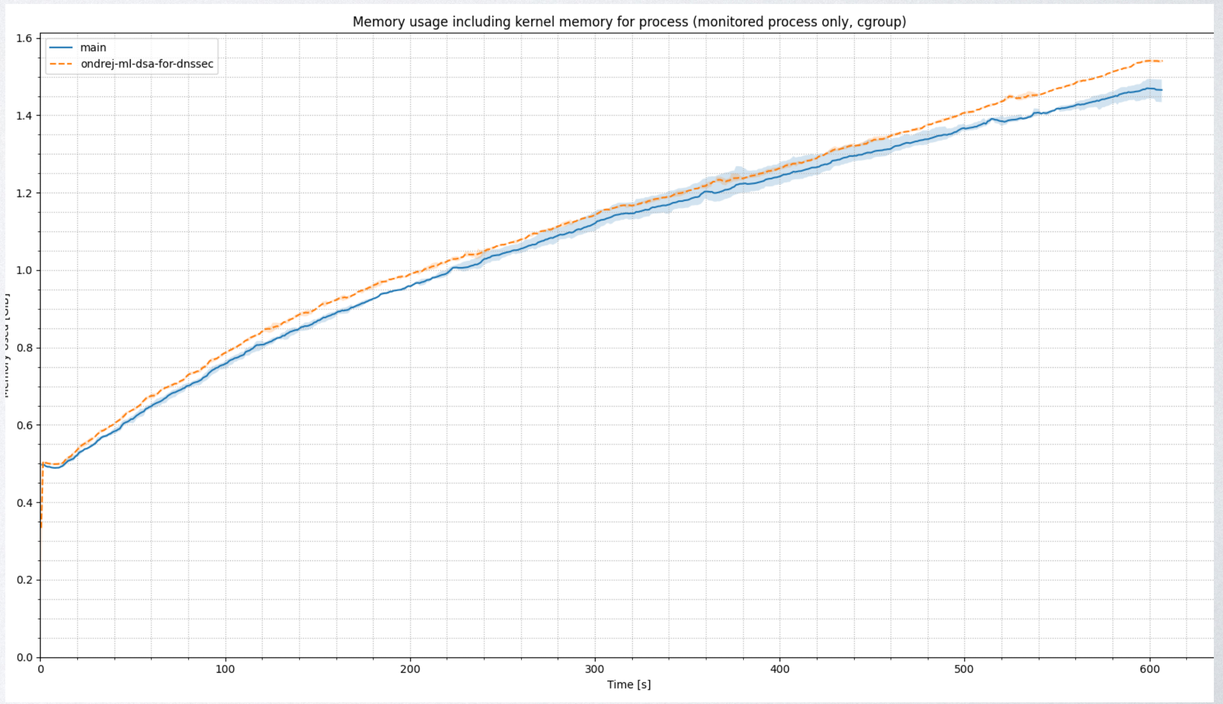

Memory usage including kernel memory for process (monitored process only, cgroup)
ondrej-pqc-main
ondrej-pqc-antrag
Time [s]

ML-DSA


Latency, all-groups, since test time 0 until 600
main (198 k QPS)
ondrej-ml-dsa-for-dnssec (198 k QPS)
Slowest percentile


Latency, all-groups, since test time 300 until 600
main (198 k QPS)
ondrej-ml-dsa-for-dnssec (198 k QPS)
Slowest percentile


CPUs usage (monitored process only, cgroup)
main
ondrej-ml-dsa-for-dnssec
Time [s]


Memory usage including kernel memory for process (monitored process only, cgroup)
main
ondrej-ml-dsa-for-dnssec
Time [s]

PLÁNY DO BUDOUCNA

- Otestovat různé úrovně hierarchie DNS
- Další (odlišné) algoritmy
- Zaměřit se na vzory pseudo-náhodných subdomén (PRSD attack)
- Vynucený re-keying (nízké TTL atd.)
- Implementovat agresivní cachování NSEC3 (pomůže to? v případě opt-out moc ne)
- Využít sondy SystemTap/DTrace/LTTng k měření kryptografických operací
- Použít optimalizované implementace (AVX2)

DÍKY ZA POZORNOST