Nexthosting

Étude · Serveurs de jeu

Test de charge de serveur : combien de bots de protocole supportent Vanilla, Paper, Purpur et Fabric avec Lithium ?

Quatre logiciels de serveur pour Minecraft et trois profils JVM, mesurés avec des bots de protocole sur une VM de test

CECI N'EST PAS UN PRODUIT MINECRAFT OFFICIEL. NON APPROUVÉ PAR MOJANG OU MICROSOFT, NI ASSOCIÉ À CES DERNIERS.

Type d'étude	Mesure propre
Éditeur	Nexthosting
Auteur	Erik Friedl (ORCID: https://orcid.org/0009-0008-2784-4314), Nexthosting
Version	1.0.0 du 22/11/2026
Période	21/11/2026 au 22/11/2026
Échantillon	18 sur 18 Exécutions de mesure évalués
Licence	CC BY 4.0
Version en ligne	nexthosting.net/fr/etudes/server-haertetest-minecraft

Sommaire

1	Résultats principaux	3
2	Résumé	4
3	Résultats	5
4	Méthode	10
5	Échantillon et période	12
6	Limites de la portée des résultats	13
7	Données et vérifiabilité	14
8	Sources	19
9	Transparence	19
10	Comment citer cette étude	20
11	Modifications et corrections	20

1 Résultats principaux

80 bots

Paper avec Aikar-Flags

En moyenne sur la fenêtre de mesure, Paper est resté dans les trois répétitions, jusqu'à 80 bots, sous la limite de 50 ms p95-MSPT fixée à l'avance, et dans l'une d'elles jusqu'à 85. Avec la limite plus stricte d'au moins 19,5 TPS en moyenne, on obtient 80 dans les trois répétitions.

Preuve : laeufe.csv, Champ kapazitaet · laeufe.csv, Champ kapazitaet_tps · Source [1]

75 bots

Purpur, indiscernable de Paper pour la capacité

Purpur a atteint 75 à 80 bots. Cette plage chevauche celle de Paper (80 à 85) ; cette mesure n'établit donc aucune différence entre les deux.

Preuve : laeufe.csv, Champ kapazitaet

65 bots

Fabric avec Lithium

Fabric avec Lithium a atteint 65 à 70 bots. C'est au-dessus de Vanilla et en dessous de Paper et Purpur ; les plages ne se chevauchent dans aucun des deux sens.

Preuve : laeufe.csv, Champ kapazitaet

55 bots

Vanilla

En moyenne sur la fenêtre de mesure, Vanilla est resté, jusqu'à 55 bots, sous 50 ms p95-MSPT dans les trois répétitions. Avec au moins 19,5 TPS en moyenne sur la fenêtre de mesure, on n'obtient que 50.

Preuve : laeufe.csv, Champ kapazitaet · laeufe.csv, Champ kapazitaet_tps

16,31 TPS

La limite MSPT ne suffit pas à elle seule

À 55 bots, Vanilla était en moyenne sur la fenêtre de mesure sous 50 ms p95-MSPT, mais ne tournait qu'à 16,31 à 17,16 ticks par seconde au lieu de 20. C'est pourquoi cette étude indique, à côté de la capacité fixée à l'avance, aussi la capacité à au moins 19,5 TPS.

Preuve : daten.csv, Champ tps · Source [2]

80 bots

G1 par défaut et Aikar-Flags : aucune différence de capacité démontrée
Paper avec G1 par défaut a atteint 80 à 85 bots, avec les Aikar-Flags également 80 à 85. Pour la capacité, il n'y a aucune différence démontrée. Observation complémentaire : pour la valeur p99, les Aikar-Flags étaient nettement plus bas dans 7 des 13 paliers communs, G1 par défaut dans aucun. Les exécutions Aikar se situaient en moyenne plus tôt dans l'ordre ; une partie de l'écart peut donc provenir de la dérive au cours de la nuit de mesure (voir les limites).

Preuve : laeufe.csv, Champ kapazitaet · p99-aikar-g1.csv, Champ spieler · Source [1]

70 bots

Generational ZGC est resté plus bas
Paper avec Generational ZGC a atteint 70 bots dans les trois répétitions, donc moins qu'avec G1 par défaut ou les Aikar-Flags (80 à 85 chacun). Cela vaut pour ce profil de charge, ce heap et cette version de Java.

Preuve : laeufe.csv, Champ kapazitaet · Source [3]

18 exécutions sur 18

Toutes les exécutions valides
Les 18 exécutions ont respecté les règles fixées à l'avance. En moyenne sur un palier, le CPU steal a atteint au plus 0,061 % et les cœurs du générateur de charge au plus 5,72 %.

Preuve : laeufe.csv, Champ lauf · daten.csv, Champ steal_pct · daten.csv, Champ cpu_bots_pct

2 Résumé

> **Question de recherche:** Combien de bots de protocole simultanés Vanilla, Paper, Purpur et Fabric avec Lithium peuvent-ils supporter sur un matériel identique, et dans quelle mesure les réglages de la JVM influencent-ils le résultat ?

Nous avons soumis Vanilla, Paper, Purpur et Fabric avec Lithium à une charge dans Minecraft 26.3 sur la même machine virtuelle. Des bots de protocole se sont connectés par paliers jusqu'à ce que le serveur soit surchargé. Les 18 exécutions étaient valides selon les règles fixées à l'avance.

La capacité retenue est le nombre de bots le plus élevé jusqu'auquel le temps de tick p95 est resté, en moyenne sur la fenêtre de mesure, inférieur à 50 ms, et ce dans les trois répétitions. Paper a atteint 80 bots, Purpur 75, Fabric avec Lithium 65 et Vanilla 55. Paper et Purpur ne se distinguent pas l'un de l'autre pour la capacité fixée à l'avance ; pour toutes les autres paires de logiciels, les plages des répétitions ne se chevauchent pas.

Pour les réglages de la JVM, aucune différence de capacité démontrée n'est apparue sur Paper entre G1 par défaut et les Aikar-Flags. Generational ZGC est resté en dessous, avec 70 bots.

Le temps de tick seul ne révèle pas toute surcharge : à 55 bots, Vanilla est resté en moyenne sous 50 ms p95, mais n'a atteint que 16,31 à 17,16 ticks par seconde au lieu de 20. C'est pourquoi la valeur à

au moins 19,5 TPS en moyenne figure à côté de chaque capacité. Les chiffres valent pour cette VM et ce profil de bots et ne se transposent pas directement à de vrais joueurs.

3 Résultats

Comment lire les résultats

La capacité fixée à l'avance repose sur le temps de tick p95. La seconde capacité exige à la place que le serveur atteigne, en moyenne sur la fenêtre de mesure, au moins 19,5 ticks par seconde sur 20 et que tous les bots restent connectés. Les deux indiquent la plus petite valeur sur trois répétitions. Une différence ne compte que si les plages des répétitions ne se chevauchent pas.

Capacité des logiciels

Nombre de bots le plus élevé avec, en moyenne sur la fenêtre de mesure, moins de 50 ms p95-MSPT dans les trois répétitions, tous avec les Aikar-Flags.



Capacité des logiciels à au moins 19,5 TPS

Ajouté après coup : nombre de bots le plus élevé avec au moins 19,5 TPS en moyenne dans les trois répétitions.



Capacité de Paper par profil JVM

Nombre de bots le plus élevé avec, en moyenne sur la fenêtre de mesure, moins de 50 ms p95-MSPT dans les trois répétitions.



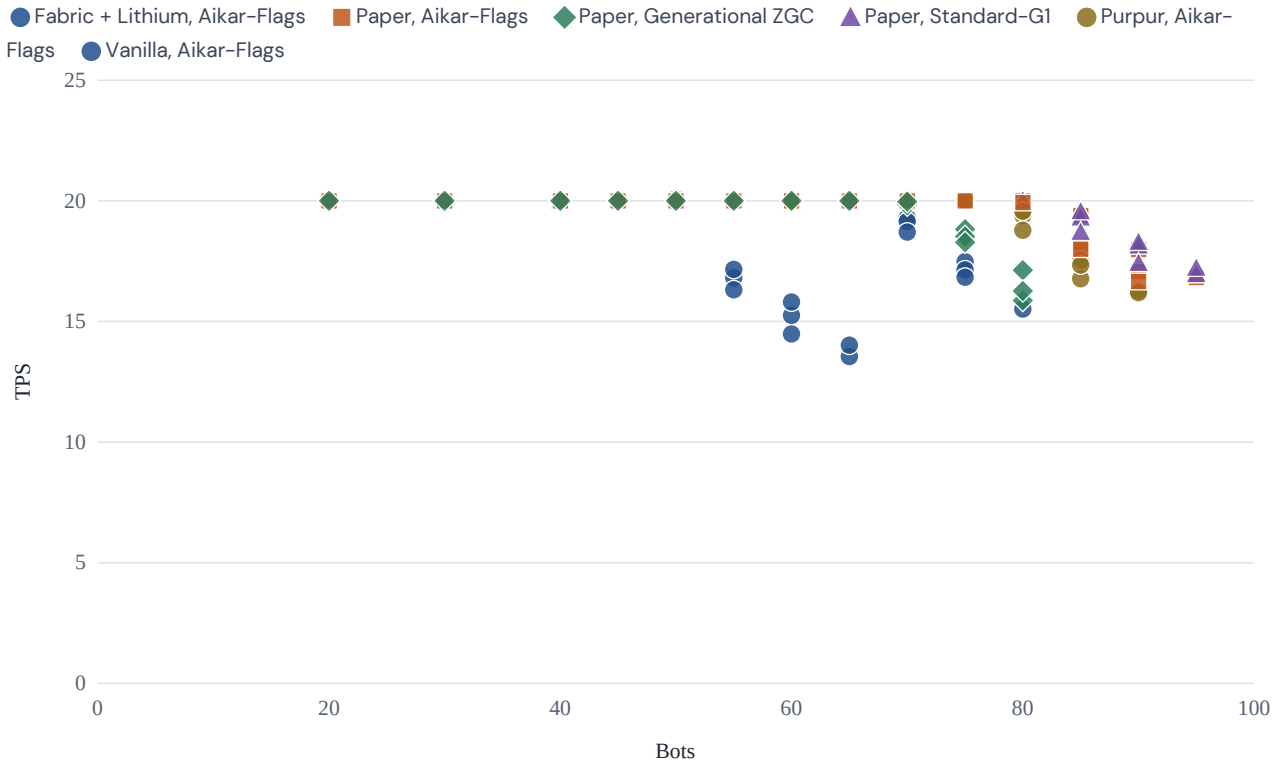
Temps de tick p95 par palier

Chaque point est un palier d'une exécution, représenté comme moyenne de la fenêtre de mesure. La limite de 50 ms s'applique à cette moyenne.



Cadence réelle par palier

Chaque point est un palier d'une exécution. L'objectif est de 20 ticks par seconde.



Toutes les combinaisons

Combinaison	Capacité	Meilleure exécution	À 19,5 TPS	Meilleure exécution (TPS)	p95 à 50, min	p95 à 50, max	p99 à 50, min	p99 à 50,
Paper, Aikar-Flags	80	85	80	80	27,8 ms	29,8 ms	31,2 ms	33,4
Paper, Standard-G1	80	85	80	85	28,3 ms	29,5 ms	34,4 ms	36,1
Purpur, Aikar-Flags	75	80	75	80	29,4 ms	31,0 ms	33,2 ms	35,0
Paper, Generational ZGC	70	70	70	70	34,1 ms	35,4 ms	39,9 ms	40,3
Fabric + Lithium, Aikar-Flags	65	70	65	65	36,1 ms	37,3 ms	43,4 ms	45,0
Vanilla, Aikar-Flags	55	55	50	50	42,7 ms	43,5 ms	50,2 ms	55,1

Les 18 exécutions

N°	Combinaison	Rép.	Capacité	TPS à ce palier	À 19,5 TPS	Valide
1	Paper, Aikar-Flags	2	85	19,40	80	Ja
2	Paper, Aikar-Flags	1	80	19,74	80	Ja
3	Paper, Generational ZGC	1	70	20,00	70	Ja
4	Purpur, Aikar-Flags	1	80	19,40	75	Ja
5	Fabric + Lithium, Aikar-Flags	2	70	19,30	65	Ja
6	Purpur, Aikar-Flags	2	75	20,00	75	Ja
7	Paper, Standard-G1	2	85	19,30	80	Ja
8	Vanilla, Aikar-Flags	3	55	16,81	50	Ja
9	Vanilla, Aikar-Flags	1	55	16,31	50	Ja
10	Fabric + Lithium, Aikar-Flags	3	65	20,00	65	Ja
11	Paper, Standard-G1	3	85	19,56	85	Ja
12	Vanilla, Aikar-Flags	2	55	17,16	50	Ja
13	Fabric + Lithium, Aikar-Flags	1	65	20,01	65	Ja
14	Paper, Standard-G1	1	80	20,01	80	Ja
15	Paper, Generational ZGC	3	70	19,82	70	Ja
16	Purpur, Aikar-Flags	3	80	19,57	80	Ja
17	Paper, Aikar-Flags	3	80	19,93	80	Ja
18	Paper, Generational ZGC	2	70	19,97	70	Ja

Comparaisons par paires

Une différence est considérée comme démontrée lorsque les plages des trois répétitions ne se chevauchent pas.

Indicateur	A	A min	A max	B	B min	B max	Démontré
kapazitaet_p95	Vanilla, Aikar-Flags	55,0	55,0	Paper, Aikar-Flags	80,0	85,0	Ja
kapazitaet_tps	Vanilla, Aikar-Flags	50,0	50,0	Paper, Aikar-Flags	80,0	80,0	Ja
p95_bei_50	Vanilla, Aikar-Flags	42,7	43,5	Paper, Aikar-Flags	27,8	29,8	Ja
p99_bei_50	Vanilla, Aikar-Flags	50,2	55,1	Paper, Aikar-Flags	31,2	33,4	Ja
kapazitaet_p95	Vanilla, Aikar-Flags	55,0	55,0	Purpur, Aikar-Flags	75,0	80,0	Ja
kapazitaet_tps	Vanilla, Aikar-Flags	50,0	50,0	Purpur, Aikar-Flags	75,0	80,0	Ja
p95_bei_50	Vanilla, Aikar-Flags	42,7	43,5	Purpur, Aikar-Flags	29,4	31,0	Ja
p99_bei_50	Vanilla, Aikar-Flags	50,2	55,1	Purpur, Aikar-Flags	33,2	35,0	Ja
kapazitaet_p95	Vanilla, Aikar-Flags	55,0	55,0	Fabric + Lithium, Aikar-Flags	65,0	70,0	Ja
kapazitaet_tps	Vanilla, Aikar-Flags	50,0	50,0	Fabric + Lithium, Aikar-Flags	65,0	65,0	Ja
p95_bei_50	Vanilla, Aikar-Flags	42,7	43,5	Fabric + Lithium, Aikar-Flags	36,1	37,3	Ja
p99_bei_50	Vanilla, Aikar-Flags	50,2	55,1	Fabric + Lithium, Aikar-Flags	43,4	45,0	Ja
kapazitaet_p95	Paper, Aikar-Flags	80,0	85,0	Purpur, Aikar-Flags	75,0	80,0	Nein
kapazitaet_tps	Paper, Aikar-Flags	80,0	80,0	Purpur, Aikar-Flags	75,0	80,0	Nein
p95_bei_50	Paper, Aikar-Flags	27,8	29,8	Purpur, Aikar-Flags	29,4	31,0	Nein
p99_bei_50	Paper, Aikar-Flags	31,2	33,4	Purpur, Aikar-Flags	33,2	35,0	Nein
kapazitaet_p95	Paper, Aikar-Flags	80,0	85,0	Fabric + Lithium, Aikar-Flags	65,0	70,0	Ja
kapazitaet_tps	Paper, Aikar-Flags	80,0	80,0	Fabric + Lithium, Aikar-Flags	65,0	65,0	Ja
p95_bei_50	Paper, Aikar-Flags	27,8	29,8	Fabric + Lithium, Aikar-Flags	36,1	37,3	Ja
p99_bei_50	Paper, Aikar-Flags	31,2	33,4	Fabric + Lithium, Aikar-Flags	43,4	45,0	Ja
kapazitaet_p95	Purpur, Aikar-Flags	75,0	80,0	Fabric + Lithium, Aikar-Flags	65,0	70,0	Ja
kapazitaet_tps	Purpur, Aikar-Flags	75,0	80,0	Fabric + Lithium, Aikar-Flags	65,0	65,0	Ja
p95_bei_50	Purpur, Aikar-Flags	29,4	31,0	Fabric + Lithium, Aikar-Flags	36,1	37,3	Ja
p99_bei_50	Purpur, Aikar-Flags	33,2	35,0	Fabric + Lithium, Aikar-Flags	43,4	45,0	Ja
kapazitaet_p95	Paper, Aikar-Flags	80,0	85,0	Paper, Standard-G1	80,0	85,0	Nein
kapazitaet_tps	Paper, Aikar-Flags	80,0	80,0	Paper, Standard-G1	80,0	85,0	Nein
p95_bei_50	Paper, Aikar-Flags	27,8	29,8	Paper, Standard-G1	28,3	29,5	Nein
p99_bei_50	Paper, Aikar-Flags	31,2	33,4	Paper, Standard-G1	34,4	36,1	Ja
kapazitaet_p95	Paper, Aikar-Flags	80,0	85,0	Paper, Generational ZGC	70,0	70,0	Ja

Indicateur	A	A min	A max	B	B min	B max	Démontré
kapazitaet_tps	Paper, Aikar-Flags	80,0	80,0	Paper, Generational ZGC	70,0	70,0	Ja
p95_bei_50	Paper, Aikar-Flags	27,8	29,8	Paper, Generational ZGC	34,1	35,4	Ja
p99_bei_50	Paper, Aikar-Flags	31,2	33,4	Paper, Generational ZGC	39,9	40,3	Ja
kapazitaet_p95	Paper, Standard-G1	80,0	85,0	Paper, Generational ZGC	70,0	70,0	Ja
kapazitaet_tps	Paper, Standard-G1	80,0	85,0	Paper, Generational ZGC	70,0	70,0	Ja
p95_bei_50	Paper, Standard-G1	28,3	29,5	Paper, Generational ZGC	34,1	35,4	Ja
p99_bei_50	Paper, Standard-G1	34,4	36,1	Paper, Generational ZGC	39,9	40,3	Ja

4 Méthode

1 **Figer les logiciels**

Tous les logiciels ont tourné sur Minecraft 26.3. Java, les logiciels de serveur et Lithium ont été vérifiés avec la somme de contrôle publiée. Il n'existe pas de somme de contrôle publiée pour le lanceur Fabric ; sa propre valeur SHA-256 y est consignée. Les server.properties étaient identiques pour tous les logiciels, les autres réglages sont restés aux valeurs par défaut de chaque logiciel. La seule autre modification concerne Paper et Purpur : la limitation des connexions par adresse IP a été désactivée, car tous les bots viennent de la même adresse de bouclage. Lithium se décrit comme un mod d'optimisation qui ne doit pas modifier la mécanique du jeu (source 4), Purpur comme un remplacement direct de Paper (source 5).

2 **Un monde commun**

Le serveur Vanilla non modifié a généré une fois, avec la graine 20261121, un monde de 31.329 chunks entièrement générés. Chaque exécution est partie d'une copie neuve de ce monde.

3 **Cœurs séparés**

Le serveur était fixé sur les cœurs 0 à 3, le générateur de charge sur les cœurs 4 à 6. Le script de mesure n'était lié à aucun cœur. Pendant la mesure, les données brutes ont été compressées toutes les 15 minutes, avec la priorité la plus basse, sur le cœur 7. L'entrée messung: 7 dans plan.json n'a pas été appliquée comme affectation de cœur. Le serveur et RCON n'étaient accessibles que sur l'adresse de bouclage, online-mode=false n'était réglé que dans ce montage isolé. L'EULA de Minecraft a été acceptée par Erik Friedl pour ces serveurs de test.

4 **Bots de protocole**

Un client léger maison parle le véritable protocole réseau de la version 26.3 : connexion, configuration, phase de jeu, keep-alive et confirmations de téléportation. Le client envoie les accusés de réception de chunks lorsqu'il reconnaît un message de lot de chunks. C'était le cas avec Vanilla et Fabric ; avec Paper et Purpur, il n'a compté aucun message de ce type dans aucune exécution. La base de données de protocole utilisée par mineflayer ne contenait pas encore la version 26.3 au moment de la mesure. Chaque bot a une graine fixe, envoie à une cadence cible de 20 paquets de position par seconde et se déplace à la vitesse de marche (0,2158 bloc par tick) entre des points de passage aléatoires. Six bots forment un groupe, les groupes sont espacés de 256 blocs. Les bots volent en mode créatif à la hauteur 120.

5 **Paliers et fenêtre de mesure**

Les paliers étaient 20, 30 et 40 bots, puis par pas de cinq. Après la connexion, 60 secondes de stabilisation et 120 secondes de fenêtre de mesure avec un échantillon toutes les 5 secondes. Une exécution s'arrêtait après deux paliers au-dessus de la limite ou en dessous de 15 TPS.

6 **Grandeurs mesurées**

MSPT p50, p95 et p99 provenaient, pour tous les logiciels, de la commande /tick query via RCON, qui évalue les 100 derniers ticks. La valeur d'un palier est la moyenne de ces blocs sur la fenêtre de mesure. Les TPS ont été calculés à partir de l'avancement du temps de jeu. S'y ajoutent la RAM du processus Java, le CPU par groupe de cœurs, le CPU steal et le nombre de bots confirmé par la liste des joueurs.

7 **Essais préliminaires avant le plan**

Avant la fixation du plan, deux exécutions pilotes (Paper, Vanilla) et de courtes exécutions d'essai de toutes les combinaisons ont eu lieu. Ensuite, la grille de paliers et les durées de mesure ont été fixées ; la limite de 50 ms était déjà arrêtée avant. Les essais préliminaires n'entrent pas dans l'évaluation.

8 **Validité**

Fixé à l'avance : CPU steal inférieur à 2 %, générateur de charge inférieur à 70 % de CPU et inférieur à 20 ms de retard des minuteries (p99), échantillons complets, tous les bots connectés tant que le serveur n'est pas surchargé, et deux bots témoins se déplacent de façon démontrable dans la fenêtre de mesure. Les exécutions invalides auraient été répétées ; il n'y en a pas eu.

9 **Capacité et différences**

Fixé à l'avance : la capacité d'une exécution est le palier le plus élevé jusqu'auquel tous les paliers restent, en moyenne sur la fenêtre de mesure, sous 50 ms p95-MSPT et tous les bots sont connectés. La capacité d'une combinaison est la plus petite valeur de ses trois exécutions. Une différence ne compte que si les plages des répétitions ne se chevauchent pas.

10 Complément ultérieur

Après la mesure, il est apparu que la cadence tombait parfois déjà sous 20 TPS alors que le temps de tick p95 restait en moyenne sous 50 ms. C'est pourquoi la capacité à au moins 19,5 TPS et les indicateurs à 50 bots ont aussi été évalués. Le palier 50 est le plus élevé où toutes les combinaisons atteignaient encore, en moyenne sur la fenêtre de mesure, au moins 19,5 TPS. Ces évaluations n'étaient pas fixées à l'avance et sont signalées comme telles dans les données.

Dispositif de mesure

Version du jeu	Minecraft 26.3 (protocole 777)
Vanilla	26.3
Paper	26.3, build 257
Purpur	26.3, build 2660
Fabric	Loader 0.20.1, Installer 1.1.2
Lithium	0.26.4 pour 26.3
Java	Temurin 25.0.5.1+2
Heap	10 GB (Xms égal à Xmx)
Profil JVM G1 par défaut	-XX:+UseG1GC
Profil JVM Aikar-Flags	Flags selon la documentation de PaperMC
Profil JVM Generational ZGC	-XX:+UseZGC
Machine virtuelle	8 vCPU sous KVM, 31,3 GiB de RAM, 100 GiB de disque dur virtuel, Debian 12
CPU signalé	QEMU Virtual CPU, AuthenticAMD ; le modèle physique ne peut pas être déterminé depuis la VM
Cœurs	Serveur 0 à 3, bots 4 à 6, sauvegarde 7, script de mesure non lié
Distance d'affichage	10 chunks, distance de simulation 10 chunks
Mode de jeu	Créatif, difficulté normale
Période de mesure	21.11.2026, 20:12 UTC jusqu'au 22.11.2026, 06:59 UTC

5 Échantillon et période

Période	21/11/2026 au 22/11/2026
Population de l’étude	18 Exécutions de mesure
Évalué	18 Exécutions de mesure

Sélection	Six combinaisons de logiciel et de profil JVM, avec trois répétitions chacune, dans un ordre aléatoire fixé à l'avance.

6 Limites de la portée des résultats

Les bots ne sont pas de vrais joueurs. Les bots volent, ne construisent pas, ne combattent pas, ne discutent pas, n'utilisent pas d'inventaire et ne déclenchent aucun circuit de redstone. De vrais joueurs génèrent une autre charge. Les nombres absolus de bots ne se transposent donc pas à un vrai serveur. Le monde avait été généré à l'avance et les bots n'ont pas quitté cette zone. La charge liée à la génération de nouveaux chunks n'est donc pas incluse.

Une VM. La mesure a été faite sur une seule machine virtuelle, avec quatre cœurs pour le serveur. Sur un autre matériel, les valeurs seront différentes, et les écarts entre les logiciels peuvent eux aussi changer.

Un profil de test. Il y avait un monde, une graine, un profil de déplacement et, hormis Lithium, aucun mod ni plugin. Les logiciels n'ont pas été optimisés individuellement. Avec une configuration adaptée ou une autre charge, l'ordre peut être différent. L'alternance jour-nuit était active. Comme toutes les exécutions suivent le même calendrier, certains paliers tombent toujours la nuit. L'influence n'a pas été isolée dans cette mesure.

Configuration par défaut différente. La configuration par défaut de Paper et Purpur contient le réglage entity-activation-range (animaux et monstres 32 blocs), absent des modèles de Vanilla et Fabric. On n'a pas vérifié si tous les logiciels ont effectué le même travail de simulation ; le nombre d'entités n'a pas été relevé.

MSPT et TPS divergent. Chez Vanilla, le temps de tick p95 à 55 bots était en moyenne sur la fenêtre de mesure inférieur à 50 ms, mais la cadence était de 16,31 à 17,16 TPS. La cause n'a pas été examinée. Pour Vanilla, la capacité fixée à l'avance est donc supérieure de 5 bots à la valeur à au moins 19,5 TPS. Pour les cinq autres combinaisons, les deux plus petites valeurs coïncident.

Moyennes plutôt que valeurs individuelles. La limite de 50 ms s'applique à la moyenne des blocs de 100 ticks sur la fenêtre de mesure. Dans 13 des 18 exécutions, au moins un bloc individuel du palier de capacité était à 50 ms ou au-dessus, au maximum à 54,2 ms. La limite de 19,5 TPS s'applique elle aussi à la moyenne de la fenêtre de mesure. Dans les paliers de la capacité à au moins 19,5 TPS, certains échantillons individuels de 5 secondes étaient en dessous, le plus bas à 17,62 TPS (Vanilla, 50 bots).

Dérive au cours de la nuit de mesure. À nombre de bots égal, le temps de tick p95 augmentait avec la position dans l'ordre des exécutions, à 50 bots d'environ 0,11 ms par exécution (évalué après coup). La cause n'a pas été examinée dans cette mesure. L'ordre était aléatoire, mais non équilibré par blocs ; la dispersion des répétitions contient donc aussi cet effet.

Indicateurs ajoutés après coup. La capacité à au moins 19,5 TPS, le palier de comparaison à 50 bots et la comparaison p99 entre Aikar-Flags et G1 par défaut n'ont été choisis qu'après examen des données. Ce sont des observations complémentaires et non des contrôles fixés à l'avance.

Grille grossière, trois répétitions. Dans la plage pertinente, les paliers sont espacés de cinq bots ; des différences plus petites ne sont pas mesurables. Trois répétitions par combinaison ne montrent la dispersion que grossièrement. Aucun test statistique n'a été calculé.

Valeurs de RAM non comparables. Les Aikar-Flags contiennent AlwaysPreTouch, qui réserve la mémoire dès le démarrage (source 1). La mémoire vive mesurée dépend donc du réglage de la JVM et ne montre pas le besoin du logiciel. Cette étude ne dit rien sur le logiciel qui nécessite le moins de RAM.

Comparaison JVM uniquement sur Paper. Les trois profils JVM n'ont été mesurés qu'avec Paper, un heap de 10 GB et Java 25. La comparaison des logiciels n'a été faite qu'avec les Aikar-Flags. Aucun des deux résultats ne se transpose sans autre à d'autres combinaisons.

Instantané. Les résultats valent pour les versions et builds indiqués. Des versions ultérieures peuvent se comporter différemment.

7 Données et vérifiabilité

Pour répéter la mesure, il faut une machine Linux avec huit cœurs et environ 32 GB de RAM comme ici, Node.js et Python 3. Les scripts attendent le dossier de travail /root/studien/haertetest, téléchargent Java et les mêmes versions des logiciels tant que ces téléchargements sont disponibles, vérifient les sommes de contrôle, génèrent le monde et exécutent le plan de mesure. La graine, l'ordre, les paliers et les règles figurent dans plan.json. Un autre matériel, d'autres builds ou une autre charge du système hôte conduisent à d'autres valeurs absolues.

Fichier	Taille	SHA-256
daten.csv Valeurs mesurées par palier (CSV)	35,6 KB	a2c2bfdfcc8ddfce2f6eac4044c15aeb ed57f4d0fab95b8a48af72c47e2f9e9e
daten.json Valeurs mesurées par palier (JSON)	94,6 KB	f798d400aba35617cf30f8c0cd2ad07c f6aace6e19e02300a0a1b822c6ac980e
laeufe.csv Résultat par exécution	2,7 KB	3a7f0af91987e982f8e95f97c60ef833 0ea9ba8a9f4688e76c64f8514b3486d1
zusammenfassung.csv Résultat par combinaison	1,1 KB	8ba8269c39e20d767c2a8c87dfb805fd 0d937bb59bd448680f23401f0e66bd88
vergleiche.csv Comparaisons par paires	4,7 KB	99575f03f6a3f2075bcd20426a18e86d 6d14ad5410618ebb101050c3fabcda9d
p99-aikar-g1.csv p99 Aikar-Flags et G1 par défaut	0,5 KB	4d0b6c017068a81642cddc339b1623f2 71d5d543abaf0ea9e4722152907e64ac

Fichier	Taille	SHA-256
proben.csv Échantillons individuels	517,9 KB	7fa08cf87e94c1e58c680ae9cec668ba 6d56da228bda6be08849d8232db27c33
plan.json Plan de mesure	5,9 KB	cc335b804f52374f6c77afda5c55dd19 dbf5fa03d595c8f3a9bb72cedd569192
messprotokoll.txt Journal de mesure	8,1 KB	f234fa1d0a16b312777dd0b93fb12f0e b13161fbdbb13125501e034837705a64
herkunft.txt Origine des logiciels	1,4 KB	5f9e921b1c4fe697d2e676b7b5777eee ae784a4c4418b211230e7ac333bb676d
liesmich.txt Lisez-moi	1,4 KB	8a444fae54933207d6f896459ac23511 c541a656d0fc2da2a26b2f69091f0935
01-einrichten.sh Installation	3,4 KB	27627241f4e7964be26e9b3ac2b3c1de 74482e323348569f6e7ba6e9290b7ea9
02-vorlagen.py Monde et modèles	4,1 KB	fc70d06d68b18d70425e0987794f06da 257659424a168433c2048577258d4cd5
03-plan.py Plan de mesure	3,5 KB	773d40d0d4dd1323c748a8c383ecf1ff b0fb91299cbea545f21144dfa083a034
gemeinsam.py Éléments communs	5,3 KB	da16b51f5b78fd55c070c1eaa9b43485 cd7fd1654ce8c47e71df87b26bcb7e20
messlauf.py Exécution de mesure	10,9 KB	395ec326a334855d03a333879da16f90 0730d0307f06065f078f79af8de21637
hauptlauf.py Mesure principale	4,6 KB	7b1602088aba30734e706e35ab894d4c 918a91eec8b0846a68a2ee392c6cec2f
bots.mjs Générateur de charge	14,3 KB	6fa44c58b3c56a31fb77fb19602cc06a ab93f08e8e8984a40bb95f2ad1064a10
welt-pruefen.py Contrôle du monde	1,5 KB	e0bb7dcc1e2327e9b916355b3e3dba55 2751a2f6c13989ffd07f10c234c0940e
auswertung.mjs Évaluation	10,0 KB	42dc3452a2061bad7ede2d84610cdf8c 8ef3adb5e7c6c7b8593c502aa2b85a33
baue-studie.mjs Construction de l'étude	39,2 KB	c0e76feab583afa11baf3bbce105203f 4a3ebcf57e66adf44380bb7854a8b90b

Champs des fichiers de données

Champ	Unité	Signification
lauf		Identifiant de l'exécution de mesure avec sa position dans l'ordre aléatoire
kombination		Identifiant composé du logiciel et du profil JVM
anzeige		Nom lisible de la combinaison

Champ	Unité	Signification
programm		Logiciel de serveur
jvm		Profil JVM
wiederholung		Numéro de la répétition, 1 à 3
versuch		Tentative au sein de la répétition, 1 est la première
reihenfolge		Position dans l'ordre aléatoire fixé
gueltig		L'exécution remplit toutes les règles de validité fixées à l'avance
lauf_gueltig		L'exécution correspondante est valide
spieler	Bots	Nombre cible de bots dans ce palier
spieler_min	Bots	Plus petit nombre de bots confirmé par la liste des joueurs dans la fenêtre de mesure
spieler_verbunden	Bots	Nombre de bots confirmé par la liste des joueurs dans cet échantillon
unter_grenze		p95-MSPT en moyenne sur la fenêtre de mesure inférieur à 50 ms et tous les bots connectés
p50_ms	ms	Médiane du temps de tick ; pour les paliers, moyenne des blocs de 100 ticks
p95_ms	ms	95e percentile du temps de tick ; pour les paliers, moyenne des blocs de 100 ticks
p95_max_ms	ms	Plus haute valeur p95 d'un bloc de 100 ticks dans la fenêtre de mesure
p99_ms	ms	99e percentile du temps de tick ; pour les paliers, moyenne des blocs de 100 ticks
mspt_mittel_ms	ms	Temps de tick moyen selon /tick query
tps	Ticks/s	Cadence réelle d'après l'avancement du temps de jeu ; pour les paliers, moyenne de la fenêtre de mesure
tps_min	Ticks/s	Cadence la plus basse d'un échantillon de 5 secondes dans la fenêtre de mesure
ram_gb	GiB	Mémoire vive occupée par le processus Java (RSS)
cpu_server_pct	%	Utilisation moyenne des quatre cœurs du serveur
cpu_bots_pct	%	Utilisation moyenne des trois cœurs du générateur de charge

Champ	Unité	Signification
steal_pct	%	CPU steal sur l'ensemble des huit cœurs
bots_verzug_p99_ms	ms	Plus haut retard p99 des minuteries dans le générateur de charge
getrennt	Bots	Bots déconnectés de façon inattendue jusqu'à ce palier
proben		Nombre d'échantillons de 5 secondes dans la fenêtre de mesure
zeit_unix	s	Instant de l'échantillon en temps Unix
erste_ueberlast	Bots	Premier palier dont la p95-MSPT moyenne dépasse la limite de 50 ms
kapazitaet	Bots	Capacité fixée à l'avance ; dans le récapitulatif, plus petite valeur des trois exécutions
tps_bei_kapazitaet	Ticks/s	Cadence moyenne dans le palier de capacité
kapazitaet_tps	Bots	Palier le plus élevé jusqu'auquel tous les paliers ont tenu au moins 19,5 TPS en moyenne sur la fenêtre de mesure ; dans le récapitulatif, plus petite valeur
kapazitaet_tps_190	Bots	Comme kapazitaet_tps, avec 19,0 TPS comme limite
kapazitaet_tps_198	Bots	Comme kapazitaet_tps, avec 19,8 TPS comme limite
p95_bei_50_ms	ms	p95-MSPT dans le palier à 50 bots
p99_bei_50_ms	ms	p99-MSPT dans le palier à 50 bots
tps_bei_50	Ticks/s	Cadence dans le palier à 50 bots
stufen_gemessen		Nombre de paliers mesurés de l'exécution
ungueltig_grund		Motif pour les exécutions invalides, sinon vide
im_programmvergleich		La combinaison fait partie de la comparaison des logiciels
im_jvm_vergleich		La combinaison fait partie de la comparaison des profils JVM
laeufe_gueltig		Exécutions valides de la combinaison
laeufe_ungueltig		Exécutions invalides de la combinaison
kapazitaet_median	Bots	Médiane de la capacité sur les exécutions
kapazitaet_max	Bots	Plus grande valeur de capacité des exécutions
kapazitaet_tps_median	Bots	Médiane de kapazitaet_tps
kapazitaet_tps_max	Bots	Plus grande valeur de kapazitaet_tps

Champ	Unité	Signification
tps_bei_kapazitaet_min	Ticks/s	Cadence la plus basse à la capacité sur les exécutions
p95_bei_50_min_ms	ms	Plus petite valeur p95 à 50 bots
p95_bei_50_max_ms	ms	Plus grande valeur p95 à 50 bots
p99_bei_50_min_ms	ms	Plus petite valeur p99 à 50 bots
p99_bei_50_max_ms	ms	Plus grande valeur p99 à 50 bots
ram_bei_50_gb	GiB	RSS moyen à 50 bots
cpu_bei_50_pct	%	Utilisation moyenne des cœurs du serveur à 50 bots
vergleich		Comparaison des logiciels ou comparaison JVM
kennzahl		Indicateur comparé
regel		Indique si l'indicateur a été fixé à l'avance ou ajouté après coup
a		Première combinaison
a_min		Plus petite valeur de la première combinaison
a_max		Plus grande valeur de la première combinaison
b		Seconde combinaison
b_min		Plus petite valeur de la seconde combinaison
b_max		Plus grande valeur de la seconde combinaison
unterschied_belegt		Les plages des répétitions ne se chevauchent pas
besser		Combinaison avec la meilleure valeur, uniquement en cas de différence démontrée
aikar_min	ms	Plus petite valeur p99 avec les Aikar-Flags dans le palier
aikar_max	ms	Plus grande valeur p99 avec les Aikar-Flags dans le palier
g1_min	ms	Plus petite valeur p99 avec G1 par défaut dans le palier
g1_max	ms	Plus grande valeur p99 avec G1 par défaut dans le palier
aikar_niedriger		Les trois valeurs Aikar sont toutes inférieures aux trois valeurs G1
g1_niedriger		Les trois valeurs G1 sont toutes inférieures aux trois valeurs Aikar

8 Sources

[1] PaperMC (2026): Aikar's flags.
https://docs.papermc.io/paper/aikars-flags/
Consulté le 21/11/2026

[2] Minecraft Wiki (2026): Commands/tick.
https://minecraft.wiki/w/Commands/tick
Consulté le 21/11/2026

[3] OpenJDK: JEP 490: ZGC: Remove the Non-Generational Mode.
https://openjdk.org/jeps/490
Consulté le 21/11/2026

[4] Lithium-Projekt auf Modrinth (2026): Lithium.
https://modrinth.com/mod/lithium
Consulté le 21/11/2026

[5] PurpurMC: Purpur Documentation.
https://purpurmc.org/docs/purpur/
Consulté le 21/11/2026

9 Transparence

Intérêt propre

Nexthosting vend elle-même des serveurs de jeu, dont des serveurs pour Minecraft, et a donc un intérêt économique dans ce sujet. Aucun des logiciels testés ne provient de Nexthosting. La mesure a été réalisée sur une VM de test commandée par Nexthosting. L'étude n'évalue aucune offre d'hébergeurs. Nexthosting est responsable de cette étude, contact : support@nexthosting.net.

Auteur

Erik Friedl, Nexthosting. L’éditeur est Nexthosting.

Licence

CC BY 4.0: https://creativecommons.org/licenses/by/4.0/deed.fr

10 Comment citer cette étude

APA

Friedl, E. & Nexthosting. (2026). Test de charge de serveur : combien de bots de protocole supportent Vanilla, Paper, Purpur et Fabric avec Lithium ? (Version 1.0.0) [Rapport]. Nexthosting. https://nexthosting.net/fr/etudes/server-haertetest-minecraft

BibTeX

```
@techreport{nexthosting2026serverhaertetestminecraft,
  author     = {Erik Friedl and {Nexthosting}},
  title      = {Test de charge de serveur : combien de bots de protocole supportent Vanilla, Paper, Purpur et
Fabric avec Lithium ?},
  institution= {Nexthosting},
  type       = {Étude},
  year       = {2026},
  version    = {1.0.0},
  url        = {https://nexthosting.net/fr/etudes/server-haertetest-minecraft},
  urldate    = {2026-11-22},
  language   = {french}
}
```

11 Modifications et corrections

Version	Date	Type	Modification
1.0.0	22/11/2026	Version	Première version.