Nexthosting

Studio · Server di gioco

Stress test dei server: quanti bot di protocollo reggono Vanilla, Paper, Purpur e Fabric con Lithium?

Quattro programmi server per Minecraft e tre profili JVM, misurati con bot di protocollo su una VM di test

NON È UN PRODOTTO MINECRAFT UFFICIALE. NON APPROVATO DA MOJANG O MICROSOFT, NÉ ASSOCIATO A ESSE.

Tipo di studio	Misurazione propria
Editore	Nexthosting
Autore	Erik Friedl (ORCID: https://orcid.org/0009-0008-2784-4314), Nexthosting
Versione	1.0.0 del 22/11/2026
Periodo	21/11/2026 al 22/11/2026
Campione	18 di 18 Esecuzioni di misura valutati
Licenza	CC BY 4.0
Versione online	nexthosting.net/it/studi/server-haertetest-minecraft

Indice

1	Risultati principali	3
2	Riepilogo	4
3	Risultati	5
4	Metodo	10
5	Campione e periodo	12
6	Limiti della portata dei risultati	13
7	Dati e verificabilità	14
8	Fonti	18
9	Trasparenza	19
10	Come citare questo studio	19
11	Modifiche e correzioni	20

1 Risultati principali

80 bot

Paper con i flag di Aikar

In media nella finestra di misura, in tutte e tre le ripetizioni Paper è rimasto fino a 80 bot sotto il limite stabilito in anticipo di 50 ms di MSPT p95, in una di esse fino a 85. Con il limite più severo di almeno 19,5 TPS in media, il valore è stato 80 in tutte e tre le ripetizioni.

Prova: laeufe.csv, Campo kapazitaet · laeufe.csv, Campo kapazitaet_tps · Fonte [1]

75 bot

Purpur, non distinguibile da Paper per capacità

Purpur ha raggiunto da 75 a 80 bot. L'intervallo si sovrappone a quello di Paper (da 80 a 85), quindi questa misurazione non evidenzia alcuna differenza tra i due.

Prova: laeufe.csv, Campo kapazitaet

65 bot

Fabric con Lithium

Fabric con Lithium ha raggiunto da 65 a 70 bot. Il valore è superiore a Vanilla e inferiore a Paper e Purpur; gli intervalli non si sovrappongono in nessuna delle due direzioni.

Prova: laeufe.csv, Campo kapazitaet

55 bot

Vanilla

In media nella finestra di misura, in tutte e tre le ripetizioni Vanilla è rimasto fino a 55 bot sotto 50 ms di MSPT p95. Con almeno 19,5 TPS in media nella finestra di misura sono stati solo 50.

Prova: laeufe.csv, Campo kapazitaet · laeufe.csv, Campo kapazitaet_tps

16,31 TPS

Il solo limite MSPT non basta

Con 55 bot Vanilla era in media nella finestra di misura sotto 50 ms di MSPT p95, ma girava solo a una frequenza da 16,31 a 17,16 tick al secondo invece di 20. Per questo lo studio riporta, accanto alla capacità stabilita in anticipo, anche la capacità con almeno 19,5 TPS.

Prova: daten.csv, Campo tps · Fonte [2]

80 bot

G1 standard e flag di Aikar: nessuna differenza di capacità documentata
Paper con G1 standard ha raggiunto da 80 a 85 bot, e altrettanto con i flag di Aikar, da 80 a 85. Per la capacità non c'è alcuna differenza documentata. Osservazione aggiuntiva: nel valore p99, in 7 di 13 livelli comuni i flag di Aikar sono risultati nettamente più bassi, G1 standard in nessuno. Le esecuzioni con Aikar si collocavano in media prima nell'ordine, quindi una parte della distanza può derivare dalla deriva nel corso della notte di misura (vedi limiti).
Prova: laeufe.csv, Campo kapazitaet · p99-aikar-g1.csv, Campo spieler · Fonte [1]

70 bot

Generational ZGC è rimasto più basso
Paper con Generational ZGC ha raggiunto 70 bot in tutte e tre le ripetizioni, quindi meno che con G1 standard o con i flag di Aikar (80 a 85 ciascuno). Questo vale per questo profilo di carico, questo heap e questa versione di Java.
Prova: laeufe.csv, Campo kapazitaet · Fonte [3]

18 esecuzioni su 18

Tutte le esecuzioni valide
Tutte le 18 esecuzioni hanno rispettato le regole stabilite in anticipo. In media su un livello, il CPU steal ha raggiunto al massimo lo 0,061 %, i core del generatore di carico al massimo il 5,72 %.
Prova: laeufe.csv, Campo lauf · daten.csv, Campo steal_pct · daten.csv, Campo cpu_bots_pct

2 Riepilogo

> **Domanda di ricerca:** Quanti bot di protocollo simultanei reggono Vanilla, Paper, Purpur e Fabric con Lithium a parità di hardware, e quanto influenzano il risultato le impostazioni della JVM?

Abbiamo messo sotto carico Vanilla, Paper, Purpur e Fabric con Lithium in Minecraft 26.3 sulla stessa macchina virtuale. I bot di protocollo si univano a livelli successivi fino al sovraccarico del server. Tutte le 18 esecuzioni sono risultate valide secondo le regole stabilite in anticipo.

Come capacità conta il numero massimo di bot fino al quale il tick time p95 è rimasto in media sotto 50 ms nella finestra di misura, in tutte e tre le ripetizioni. Paper ha raggiunto 80 bot, Purpur 75, Fabric con Lithium 65 e Vanilla 55. Con la capacità stabilita in anticipo, Paper e Purpur non si distinguono tra loro; in tutte le altre coppie di programmi gli intervalli delle ripetizioni non si sovrappongono.

Per quanto riguarda le impostazioni della JVM, con Paper non è emersa alcuna differenza di capacità documentata tra G1 standard e i flag di Aikar. Generational ZGC è rimasto sotto, con 70 bot.

Il solo tick time non rivela ogni sovraccarico: con 55 bot Vanilla è rimasto in media sotto 50 ms p95, ma ha raggiunto solo da 16,31 a 17,16 tick al secondo invece di 20. Per questo, accanto a ogni capacità, è riportato anche il valore con almeno 19,5 TPS in media. I numeri valgono per questa VM

e per questo profilo di bot e non possono essere trasferiti direttamente a giocatori reali.

3 Risultati

Come leggere i risultati

La capacità stabilita in anticipo si basa sul tick time p95. La seconda capacità richiede invece che il server regga in media nella finestra di misura almeno 19,5 su 20 tick al secondo e che tutti i bot restino connessi. Entrambe riportano il valore più basso tra tre ripetizioni. Una differenza conta solo se gli intervalli delle ripetizioni non si sovrappongono.

Capacità dei programmi

Numero massimo di bot con in media nella finestra di misura meno di 50 ms di MSPT p95 in tutte e tre le ripetizioni, tutti con i flag di Aikar.



Capacità dei programmi con almeno 19,5 TPS

Aggiunto successivamente: numero massimo di bot con in media almeno 19,5 TPS in tutte e tre le ripetizioni.



Capacità di Paper per profilo JVM

Numero massimo di bot con in media nella finestra di misura meno di 50 ms di MSPT p95 in tutte e tre le ripetizioni.



Tick time p95 per livello

Ogni punto è un livello di un'esecuzione, rappresentato come media della finestra di misura. Il limite di 50 ms vale per questa media.



Frequenza di tick effettiva per livello

Ogni punto è un livello di un'esecuzione. L'obiettivo è di 20 tick al secondo.



Tutte le combinazioni

Combinazione	Capacità	Esecuzione massima	A 19,5 TPS	Esecuzione massima (TPS)	p95 a 50, min	p95 a 50, max	p99 a 50, min	p99 a 5
Paper, Aikar-Flags	80	85	80	80	27,8 ms	29,8 ms	31,2 ms	3
Paper, Standard-G1	80	85	80	85	28,3 ms	29,5 ms	34,4 ms	
Purpur, Aikar-Flags	75	80	75	80	29,4 ms	31,0 ms	33,2 ms	3
Paper, Generational ZGC	70	70	70	70	34,1 ms	35,4 ms	39,9 ms	4
Fabric + Lithium, Aikar-Flags	65	70	65	65	36,1 ms	37,3 ms	43,4 ms	4
Vanilla, Aikar-Flags	55	55	50	50	42,7 ms	43,5 ms	50,2 ms	

Tutte le 18 esecuzioni

N.	Combinazione	Rip.	Capacità	TPS Iì	A 19,5 TPS	Valida
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

Confronti a coppie

Una differenza è considerata documentata se gli intervalli delle tre ripetizioni non si sovrappongono.

Indicatore	A	A min	A max	B	B min	B max	Documentata
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

Indicatore	A	A min	A max	B	B min	B max	Documentata
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 Metodo

1 **Fissare i programmi**

Tutti i programmi giravano su Minecraft 26.3. Java, i programmi server e Lithium sono stati verificati con il checksum pubblicato. Per l'avviatore di Fabric non esiste un checksum pubblicato, quindi è registrato il valore SHA-256 calcolato in proprio. I server.properties erano identici per tutti i programmi, le altre impostazioni sono rimaste ai valori predefiniti del rispettivo programma. L'unica altra modifica riguarda Paper e Purpur: la limitazione delle connessioni per indirizzo IP è stata disattivata, perché tutti i bot provengono dallo stesso indirizzo loopback. Lithium si descrive come una mod di ottimizzazione che non dovrebbe alterare la meccanica di gioco (fonte 4), Purpur come sostituto diretto di Paper (fonte 5).

2 **Un mondo comune**

Il server Vanilla non modificato ha generato una sola volta, con seed 20261121, un mondo di 31.329 chunk completamente generati. Ogni esecuzione è partita da una copia nuova di questo mondo.

3 **Core separati**

Il server girava vincolato ai core da 0 a 3, il generatore di carico ai core da 4 a 6. Lo script di misura non era vincolato a nessun core. Durante la misurazione, i dati grezzi venivano compressi ogni 15 minuti con la priorità più bassa sul core 7. La voce messung: 7 in plan.json non è stata applicata come vincolo di core. Server e RCON erano raggiungibili solo sull'indirizzo loopback, online-mode=false era impostato solo in questa configurazione isolata. L'EULA di Minecraft è stata accettata da Erik Friedl per questi server di test.

4 **Bot di protocollo**

Un client leggero sviluppato in proprio parla il vero protocollo di rete della versione 26.3: accesso, configurazione, fase di gioco, keep-alive e conferme di teletrasporto. Il client invia le conferme dei chunk quando riconosce un messaggio di chunk batch. Questo è avvenuto con Vanilla e Fabric; con Paper e Purpur non ne ha contato nessuno in alcuna esecuzione. Il database di protocollo usato da mineflayer non conteneva ancora la versione 26.3 al momento della misurazione. Ogni bot ha un seed fisso, invia con una cadenza nominale di 20 pacchetti di posizione al secondo e si muove a velocità di camminata (0,2158 blocchi per tick) tra waypoint casuali. Sei bot alla volta formano un gruppo, e i gruppi distano 256 blocchi l'uno dall'altro. I bot volano in modalità creativa a quota 120.

5 **Livelli e finestra di misura**

I livelli erano 20, 30 e 40 bot, poi a incrementi di cinque. Dopo l'ingresso seguivano 60 secondi di assestamento e 120 secondi di finestra di misura con un campione ogni 5 secondi. Un'esecuzione terminava dopo due livelli oltre il limite o sotto 15 TPS.

6 **Grandezze misurate**

MSPT p50, p95 e p99 provenivano, per tutti i programmi, dal comando /tick query tramite RCON, che valuta gli ultimi 100 tick. Il valore del livello è la media di questi blocchi nella finestra di misura. I TPS sono stati calcolati dall'avanzamento del tempo di gioco. A questi si aggiungevano la RAM del processo Java, la CPU per gruppo di core, il CPU steal e il numero di bot confermato tramite l'elenco dei giocatori.

7 **Prove preliminari prima del piano**

Prima di definire il piano sono state eseguite due esecuzioni pilota (Paper, Vanilla) e brevi prove di tutte le combinazioni. Solo dopo sono stati fissati la griglia dei livelli e i tempi di misura; il limite di 50 ms era già stabilito prima. Le prove preliminari non confluiscono nella valutazione.

8 **Validità**

Stabilito in anticipo: CPU steal sotto il 2 %, generatore di carico sotto il 70 % di CPU e sotto 20 ms di ritardo dei timer (p99), campioni completi, tutti i bot connessi finché il server non è sovraccarico, e due bot campione che si muovono in modo dimostrabile nella finestra di misura. Le esecuzioni non valide sarebbero state ripetute; non ce n'è stata nessuna.

9 **Capacità e differenze**

Stabilito in anticipo: la capacità di un'esecuzione è il livello più alto fino al quale tutti i livelli restano in media nella finestra di misura sotto 50 ms di MSPT p95 e tutti i bot sono connessi. La capacità di una combinazione è il valore più basso delle sue tre esecuzioni. Una differenza vale solo se gli intervalli delle ripetizioni non si sovrappongono.

10 Integrazione successiva

Dopo la misurazione è emerso che la frequenza di tick scendeva in parte già sotto 20 TPS mentre il tick time p95 era in media ancora sotto 50 ms. Per questo sono state valutate in aggiunta la capacità con almeno 19,5 TPS e gli indicatori a 50 bot. Il livello 50 è il più alto in cui tutte le combinazioni raggiungevano ancora in media nella finestra di misura almeno 19,5 TPS. Queste valutazioni non erano stabilite in anticipo e sono contrassegnate come tali nei dati.

Configurazione della misurazione

Versione di gioco	Minecraft 26.3 (protocollo 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 per 26.3
Java	Temurin 25.0.5.1+2
Heap	10 GB (Xms uguale a Xmx)
Profilo JVM G1 standard	-XX:+UseG1GC
Profilo JVM flag di Aikar	Flag secondo la documentazione di PaperMC
Profilo JVM Generational ZGC	-XX:+UseZGC
Macchina virtuale	8 vCPU su KVM, 31,3 GiB di RAM, 100 GiB di disco virtuale, Debian 12
CPU segnalata	QEMU Virtual CPU, AuthenticAMD; il modello fisico non è determinabile dall'interno della VM
Core	Server da 0 a 3, bot da 4 a 6, backup 7, script di misura non vincolato
Distanza di visualizzazione	10 chunk, distanza di simulazione 10 chunk
Modalità di gioco	Creativa, difficoltà normale
Periodo di misura	21.11.2026, 20:12 UTC fino al 22.11.2026, 06:59 UTC

5 Campione e periodo

Periodo	21/11/2026 al 22/11/2026
Popolazione dello studio	18 Esecuzioni di misura
Valutato	18 Esecuzioni di misura

Selezione	Sei combinazioni di programma e profilo JVM con tre ripetizioni ciascuna, in un ordine casuale stabilito in anticipo.

6 Limiti della portata dei risultati

I bot non sono giocatori. I bot volano, non costruiscono, non combattono, non scrivono in chat, non usano l'inventario e non attivano circuiti di redstone. I giocatori reali generano un carico diverso. I numeri assoluti di bot non possono quindi essere trasferiti a un server reale. Il mondo era stato generato in anticipo e i bot non hanno lasciato quest'area. Il carico dovuto alla generazione di nuovi chunk non è quindi incluso.

Una VM. La misurazione è stata eseguita su una sola macchina virtuale con quattro core per il server. Su hardware diverso i valori risulteranno diversi, e anche le distanze tra i programmi possono cambiare.

Un profilo di test. C'erano un mondo, un seed, un profilo di movimento e, a parte Lithium, nessuna mod o plugin. I programmi non sono stati ottimizzati singolarmente. Con una configurazione adattata o un carico diverso l'ordine può risultare diverso. Il ciclo giorno-notte era attivo. Poiché tutte le esecuzioni seguono lo stesso calendario, determinati livelli cadono sempre di notte. L'influenza non è stata rilevata separatamente in questa misurazione.

Configurazione predefinita diversa. La configurazione predefinita di Paper e Purpur contiene l'impostazione entity-activation-range (animali e mostri 32 blocchi), che manca nei modelli di Vanilla e Fabric. Non è stato verificato se tutti i programmi abbiano svolto lo stesso lavoro di simulazione; il numero di entità non è stato rilevato.

MSPT e TPS divergono. Con Vanilla il tick time p95 a 55 bot era in media nella finestra di misura sotto 50 ms, mentre la frequenza di tick andava da 16,31 a 17,16 TPS. La causa non è stata indagata. Con Vanilla la capacità stabilita in anticipo è quindi superiore di 5 bot al valore con almeno 19,5 TPS. Nelle altre cinque combinazioni i due valori minimi coincidono.

Medie invece di valori singoli. Il limite di 50 ms vale per la media dei blocchi da 100 tick nella finestra di misura. In 13 esecuzioni su 18, nel livello di capacità almeno un singolo blocco era a 50 ms o oltre, al massimo a 54,2 ms. Anche il limite di 19,5 TPS vale per la media della finestra di misura. Nei livelli della capacità con almeno 19,5 TPS, singoli campioni da 5 secondi si trovavano al di sotto, il più basso a 17,62 TPS (Vanilla, 50 bot).

Deriva nel corso della notte di misura. A parità di numero di bot, il tick time p95 è aumentato con la posizione nell'ordine delle esecuzioni, con 50 bot di circa 0,11 ms per esecuzione (valutato a posteriori). La causa non è stata indagata in questa misurazione. L'ordine era casuale ma non bilanciato a blocchi, quindi la dispersione delle ripetizioni comprende anche questo effetto.

Indicatori successivi. La capacità con almeno 19,5 TPS, il livello di confronto con 50 bot e il confronto p99 tra i flag di Aikar e G1 standard sono stati scelti solo dopo aver visto i dati. Sono osservazioni integrative e non verifiche stabilite in anticipo.

Griglia grossolana, tre ripetizioni. Nell'intervallo rilevante i livelli distano cinque bot l'uno dall'altro; differenze più piccole non sono misurabili. Tre ripetizioni per combinazione mostrano la dispersione solo in modo approssimativo. Non è stato calcolato alcun test statistico.

Valori di RAM non confrontabili. I flag di Aikar contengono AlwaysPreTouch, che riserva la memoria già all'avvio (fonte 1). La memoria misurata dipende quindi dall'impostazione della JVM e non mostra il fabbisogno del programma. Questo studio non fa alcuna affermazione su quale programma richieda meno RAM.

Confronto JVM solo con Paper. I tre profili JVM sono stati misurati solo con Paper, un heap di 10 GB e Java 25. Il confronto tra i programmi è stato eseguito solo con i flag di Aikar. Entrambi i risultati non sono trasferibili senza ulteriori verifiche ad altre combinazioni.

Istantanea. I risultati valgono per le versioni e le build indicate. Versioni successive possono comportarsi in modo diverso.

---

7 Dati e verificabilità

Per ripetere la misurazione servono una macchina Linux con otto core e circa 32 GB di RAM come qui, Node.js e Python 3. Gli script si aspettano la cartella di lavoro /root/studien/haertetest, scaricano Java e le stesse versioni dei programmi finché i download sono disponibili, verificano i checksum, generano il mondo ed eseguono il piano di misurazione. Seed, ordine, livelli e regole si trovano in plan.json. Hardware diverso, build diverse o un carico diverso del sistema host portano a valori assoluti diversi.

File	Dimensione	SHA-256
daten.csv Valori misurati per livello (CSV)	35,6 KB	a2c2bfdfcc8ddfce2f6eac4044c15aeb ed57f4d0fab95b8a48af72c47e2f9e9e
daten.json Valori misurati per livello (JSON)	94,6 KB	f798d400aba35617cf30f8c0cd2ad07c f6aace6e19e02300a0a1b822c6ac980e
laeufe.csv Risultato per esecuzione	2,7 KB	3a7f0af91987e982f8e95f97c60ef833 0ea9ba8a9f4688e76c64f8514b3486d1
zusammenfassung.csv Risultato per combinazione	1,1 KB	8ba8269c39e20d767c2a8c87dfb805fd 0d937bb59bd448680f23401f0e66bd88
vergleiche.csv Confronti a coppie	4,7 KB	99575f03f6a3f2075bcd20426a18e86d 6d14ad5410618ebb101050c3fabcda9d
p99-aikar-g1.csv p99 flag di Aikar e G1 standard	0,5 KB	4d0b6c017068a81642cddc339b1623f2 71d5d543abaf0ea9e4722152907e64ac

File	Dimensione	SHA-256
proben.csv Campioni singoli	517,9 KB	7fa08cf87e94c1e58c680ae9cec668ba 6d56da228bda6be08849d8232db27c33
plan.json Piano di misurazione	5,9 KB	cc335b804f52374f6c77afda5c55dd19 dbf5fa03d595c8f3a9bb72cedd569192
messprotokoll.txt Registro di misurazione	8,1 KB	f234fa1d0a16b312777dd0b93fb12f0e b13161fbdbb13125501e034837705a64
herkunft.txt Origine dei programmi	1,4 KB	5f9e921b1c4fe697d2e676b7b5777eee ae784a4c4418b211230e7ac333bb676d
liesmich.txt Leggimi	1,4 KB	8a444fae54933207d6f896459ac23511 c541a656d0fc2da2a26b2f69091f0935
01-einrichten.sh Configurazione	3,4 KB	27627241f4e7964be26e9b3ac2b3c1de 74482e323348569f6e7ba6e9290b7ea9
02-vorlagen.py Mondo e modelli	4,1 KB	fc70d06d68b18d70425e0987794f06da 257659424a168433c2048577258d4cd5
03-plan.py Piano di misurazione	3,5 KB	773d40d0d4dd1323c748a8c383ecf1ff b0fb91299cbea545f21144dfa083a034
gemeinsam.py Parti comuni	5,3 KB	da16b51f5b78fd55c070c1eaa9b43485 cd7fd1654ce8c47e71df87b26bcb7e20
messlauf.py Esecuzione di misura	10,9 KB	395ec326a334855d03a333879da16f90 0730d0307f06065f078f79af8de21637
hauptlauf.py Misurazione principale	4,6 KB	7b1602088aba30734e706e35ab894d4c 918a91eec8b0846a68a2ee392c6cec2f
bots.mjs Generatore di carico	14,3 KB	6fa44c58b3c56a31fb77fb19602cc06a ab93f08e8e8984a40bb95f2ad1064a10
welt-pruefen.py Verifica del mondo	1,5 KB	e0bb7dcc1e2327e9b916355b3e3dba55 2751a2f6c13989ffd07f10c234c0940e
auswertung.mjs Valutazione	10,0 KB	42dc3452a2061bad7ede2d84610cdf8c 8ef3adb5e7c6c7b8593c502aa2b85a33
baue-studie.mjs Costruzione dello studio	39,2 KB	c0e76feab583afa11baf3bbce105203f 4a3ebcf57e66adf44380bb7854a8b90b

Campi dei file di dati

Campo	Unità	Significato
lauf		Identificativo dell'esecuzione di misura con posizione nell'ordine casuale
kombination		Identificativo composto da programma e profilo JVM
anzeige		Nome leggibile della combinazione

Campo	Unità	Significato
programm		Programma server
jvm		Profilo JVM
wiederholung		Numero della ripetizione, da 1 a 3
versuch		Tentativo all'interno della ripetizione, 1 è il primo
reihenfolge		Posizione nell'ordine casuale stabilito
gueltig		L'esecuzione soddisfa tutte le regole di validità stabilite in anticipo
lauf_gueltig		L'esecuzione corrispondente è valida
spieler	Bots	Numero obiettivo di bot in questo livello
spieler_min	Bots	Numero minimo di bot confermato tramite l'elenco dei giocatori nella finestra di misura
spieler_verbunden	Bots	Numero di bot confermato tramite l'elenco dei giocatori in questo campione
unter_grenze		MSPT p95 in media nella finestra di misura sotto 50 ms e tutti i bot connessi
p50_ms	ms	Mediana del tick time; nei livelli, media dei blocchi da 100 tick
p95_ms	ms	95° percentile del tick time; nei livelli, media dei blocchi da 100 tick
p95_max_ms	ms	Valore p95 più alto di un blocco da 100 tick nella finestra di misura
p99_ms	ms	99° percentile del tick time; nei livelli, media dei blocchi da 100 tick
mspt_mittel_ms	ms	Tick time medio secondo /tick query
tps	Ticks/s	Frequenza di tick effettiva dall'avanzamento del tempo di gioco; nei livelli, media della finestra di misura
tps_min	Ticks/s	Frequenza di tick più bassa di un campione da 5 secondi nella finestra di misura
ram_gb	GiB	Memoria occupata dal processo Java (RSS)
cpu_server_pct	%	Utilizzo medio dei quattro core del server
cpu_bots_pct	%	Utilizzo medio dei tre core del generatore di carico
steal_pct	%	CPU steal su tutti gli otto core
bots_verzug_p99_ms	ms	Ritardo p99 più alto dei timer nel generatore di carico

Campo	Unità	Significato
getrennt	Bots	Bot disconnessi in modo inatteso fino a questo livello
proben		Numero di campioni da 5 secondi nella finestra di misura
zeit_unix	s	Momento del campione come Unix time
erste_ueberlast	Bots	Primo livello il cui MSPT p95 medio supera il limite di 50 ms
kapazitaet	Bots	Capacità stabilita in anticipo; nel riepilogo, valore più basso delle tre esecuzioni
tps_bei_kapazitaet	Ticks/s	Frequenza di tick media nel livello della capacità
kapazitaet_tps	Bots	Livello più alto fino al quale tutti i livelli hanno mantenuto in media nella finestra di misura almeno 19,5 TPS; nel riepilogo, valore più basso
kapazitaet_tps_190	Bots	Come kapazitaet_tps, con 19,0 TPS come limite
kapazitaet_tps_198	Bots	Come kapazitaet_tps, con 19,8 TPS come limite
p95_bei_50_ms	ms	MSPT p95 nel livello con 50 bot
p99_bei_50_ms	ms	MSPT p99 nel livello con 50 bot
tps_bei_50	Ticks/s	Frequenza di tick nel livello con 50 bot
stufen_gemessen		Numero di livelli misurati dell'esecuzione
ungueltig_grund		Motivo per le esecuzioni non valide, altrimenti vuoto
im_programmvergleich		La combinazione fa parte del confronto tra i programmi
im_jvm_vergleich		La combinazione fa parte del confronto tra i profili JVM
laeufe_gueltig		Esecuzioni valide della combinazione
laeufe_ungueltig		Esecuzioni non valide della combinazione
kapazitaet_median	Bots	Mediana della capacità sulle esecuzioni
kapazitaet_max	Bots	Valore di capacità più alto delle esecuzioni
kapazitaet_tps_median	Bots	Mediana di kapazitaet_tps
kapazitaet_tps_max	Bots	Valore più alto di kapazitaet_tps
tps_bei_kapazitaet_min	Ticks/s	Frequenza di tick più bassa alla capacità sulle esecuzioni
p95_bei_50_min_ms	ms	Valore p95 più basso a 50 bot
p95_bei_50_max_ms	ms	Valore p95 più alto a 50 bot

Campo	Unità	Significato
p99_bei_50_min_ms	ms	Valore p99 più basso a 50 bot
p99_bei_50_max_ms	ms	Valore p99 più alto a 50 bot
ram_bei_50_gb	GiB	RSS medio a 50 bot
cpu_bei_50_pct	%	Utilizzo medio dei core del server a 50 bot
vergleich		Confronto tra programmi o confronto JVM
kennzahl		Indicatore confrontato
regel		Se l'indicatore è stato stabilito in anticipo o aggiunto successivamente
a		Prima combinazione
a_min		Valore più basso della prima combinazione
a_max		Valore più alto della prima combinazione
b		Seconda combinazione
b_min		Valore più basso della seconda combinazione
b_max		Valore più alto della seconda combinazione
unterschied_belegt		Gli intervalli delle ripetizioni non si sovrappongono
besser		Combinazione con il valore più favorevole, solo in caso di differenza documentata
aikar_min	ms	Valore p99 più basso con i flag di Aikar nel livello
aikar_max	ms	Valore p99 più alto con i flag di Aikar nel livello
g1_min	ms	Valore p99 più basso con G1 standard nel livello
g1_max	ms	Valore p99 più alto con G1 standard nel livello
aikar_niedriger		Tutti e tre i valori Aikar sono inferiori a tutti e tre i valori G1
g1_niedriger		Tutti e tre i valori G1 sono inferiori a tutti e tre i valori Aikar

8 Fonti

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

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

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

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

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

9 Trasparenza

Interesse proprio

Nexthosting vende a sua volta game server, tra cui server per Minecraft, e ha quindi un interesse economico in questo argomento. Nessuno dei programmi testati proviene da Nexthosting. La misurazione è stata eseguita su una VM di test ordinata da Nexthosting. Lo studio non valuta le offerte di nessun hoster. Il responsabile di questo studio è Nexthosting, contatto: support@nexthosting.net.

Autore

Erik Friedl, Nexthosting. L’editore è Nexthosting.

Licenza

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

10 Come citare questo studio

APA

Friedl, E. & Nexthosting. (2026). Stress test dei server: quanti bot di protocollo reggono Vanilla, Paper, Purpur e Fabric con Lithium? (Versione 1.0.0) [Rapporto]. Nexthosting. https://nexthosting.net/it/studi/server-haertetest-minecraft

BibTeX

```
@techreport{nexthosting2026serverhaertetestminecraft,
  author     = {Erik Friedl and {Nexthosting}},
```

```
  title      = {Stress test dei server: quanti bot di protocollo reggono Vanilla, Paper, Purpur e Fabric con
Lithium?},
  institution= {Nexthosting},
  type       = {Studio},
  year       = {2026},
  version    = {1.0.0},
  url        = {https://nexthosting.net/it/studi/server-haertetest-minecraft},
  urldate    = {2026-11-22},
  language   = {italian}
}
```

11 Modifiche e correzioni

Versione	Data	Tipo	Modifica
1.0.0	22/11/2026	Versione	Prima versione.