Nexthosting

Estudio · Servidores de juegos

Prueba de resistencia del servidor: ¿cuántos bots de protocolo soportan Vanilla, Paper, Purpur y Fabric con Lithium?

Cuatro programas de servidor para Minecraft y tres perfiles de JVM, medidos con bots de protocolo en una VM de prueba

NO ES UN PRODUCTO OFICIAL DE MINECRAFT. NO ESTÁ APROBADO POR MOJANG NI MICROSOFT, NI ASOCIADO A ELLOS.

Tipo de estudio	Medición propia
Editor	Nexthosting
Autoría	Erik Friedl (ORCID: https://orcid.org/0009-0008-2784-4314), Nexthosting
Versión	1.0.0 del 22/11/2026
Periodo	21/11/2026 a 22/11/2026
Muestra	18 de 18 Ejecuciones de medición evaluados
Licencia	CC BY 4.0
Versión en línea	nexthosting.net/es/estudios/server-haertetest-minecraft

Contenido

1	Resultados principales	3
2	Resumen	4
3	Resultados	5
4	Método	10
5	Muestra y periodo	13
6	Límites del alcance de los resultados	13
7	Datos y verificabilidad	14
8	Fuentes	19
9	Transparencia	19
10	Cómo citar este estudio	20
11	Cambios y correcciones	20

1 Resultados principales

80 bots

Paper con Aikar flags

Paper se mantuvo, en promedio de la ventana de medición, hasta 80 bots por debajo del límite fijado de antemano de 50 ms de MSPT p95 en las tres repeticiones, y en una de ellas hasta 85. Con un mínimo de 19,5 TPS de media como límite más estricto fueron 80 en las tres repeticiones.

Prueba: laeufe.csv, Campo kapazitaet · laeufe.csv, Campo kapazitaet_tps · Fuente [1]

75 bots

Purpur, no distinguible de la capacidad de Paper

Purpur alcanzó entre 75 y 80 bots. El rango se solapa con el de Paper (80 a 85), por lo que esta medición no muestra ninguna diferencia entre ambos.

Prueba: laeufe.csv, Campo kapazitaet

65 bots

Fabric con Lithium

Fabric con Lithium alcanzó entre 65 y 70 bots. Eso queda por encima de Vanilla y por debajo de Paper y Purpur; los rangos no se solapan en ninguno de los dos sentidos.

Prueba: laeufe.csv, Campo kapazitaet

55 bots

Vanilla

Vanilla se mantuvo, en promedio de la ventana de medición, hasta 55 bots por debajo de 50 ms de MSPT p95 en las tres repeticiones. Con un mínimo de 19,5 TPS de media de la ventana de medición fueron solo 50.

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

16,31 TPS

El límite de MSPT por sí solo no basta

Con 55 bots, Vanilla estuvo, en promedio de la ventana de medición, por debajo de 50 ms de MSPT p95, pero funcionó solo a entre 16,31 y 17,16 ticks por segundo en lugar de 20. Por eso este estudio indica, junto a la capacidad fijada de antemano, también la capacidad con un mínimo de 19,5 TPS.

Prueba: daten.csv, Campo tps · Fuente [2]

80 bots

G1 estándar y Aikar flags: sin diferencia de capacidad demostrada
Paper con G1 estándar alcanzó entre 80 y 85 bots, y con Aikar flags también entre 80 y 85. En cuanto a la capacidad no hay ninguna diferencia demostrada. Observación complementaria: en el valor p99, las Aikar flags quedaron más bajas de forma separada en 7 de 13 niveles comunes, y G1 estándar en ninguno. Las ejecuciones con Aikar estuvieron, en promedio, antes en el orden, por lo que una parte de la diferencia puede proceder de la deriva a lo largo de la noche de medición (véanse los límites).

Prueba: laeufe.csv, Campo kapazitaet · p99-aikar-g1.csv, Campo spieler · Fuente [1]

70 bots

Generational ZGC quedó más bajo
Paper con Generational ZGC alcanzó 70 bots en las tres repeticiones y, por tanto, menos que con G1 estándar o con Aikar flags (80 a 85 en ambos casos). Esto vale para este perfil de carga, este heap y esta versión de Java.

Prueba: laeufe.csv, Campo kapazitaet · Fuente [3]

18 de 18 ejecuciones

Todas las ejecuciones válidas
Las 18 ejecuciones cumplieron las reglas fijadas de antemano. En promedio de un nivel, el CPU steal alcanzó como máximo 0,061 %, y los núcleos del generador de carga como máximo 5,72 %.

Prueba: laeufe.csv, Campo lauf · daten.csv, Campo steal_pct · daten.csv, Campo cpu_bots_pct

2 Resumen

> **Pregunta de investigación:** ¿Cuántos bots de protocolo simultáneos soportan Vanilla, Paper, Purpur y Fabric con Lithium con el mismo hardware, y hasta qué punto influyen los ajustes de la JVM en el resultado?

Sometimos a carga Vanilla, Paper, Purpur y Fabric con Lithium en Minecraft 26.3 sobre la misma máquina virtual. Los bots de protocolo se fueron incorporando por niveles hasta que el servidor se sobrecargó. Las 18 ejecuciones fueron válidas según las reglas fijadas de antemano.

Como capacidad cuenta el mayor número de bots hasta el cual el tiempo de tick p95 se mantuvo, en promedio de la ventana de medición, por debajo de 50 ms, y esto en las tres repeticiones. Paper alcanzó 80 bots, Purpur 75, Fabric con Lithium 65 y Vanilla 55. Paper y Purpur no se pueden distinguir con la capacidad fijada de antemano; en todos los demás pares de programas los rangos de las repeticiones no se solapan.

En cuanto a los ajustes de la JVM, en Paper no se mostró ninguna diferencia de capacidad demostrada entre G1 estándar y las Aikar flags. Generational ZGC quedó por debajo con 70 bots.

El tiempo de tick por sí solo no refleja toda sobrecarga: Vanilla se mantuvo con 55 bots, en promedio, por debajo de 50 ms p95, pero solo alcanzó entre 16,31 y 17,16 ticks por segundo en lugar de 20. Por eso, junto a cada capacidad figura también el valor con un mínimo de 19,5 TPS de media. Las cifras son válidas para esta VM y este perfil de bots y no se pueden trasladar directamente a jugadores reales.

3 Resultados

Cómo leer los resultados

La capacidad fijada de antemano se basa en el tiempo de tick p95. La segunda capacidad exige, en cambio, que el servidor alcance, en promedio de la ventana de medición, al menos 19,5 de 20 ticks por segundo y que todos los bots sigan conectados. Ambas indican el menor valor de tres repeticiones. Una diferencia solo cuenta si los rangos de las repeticiones no se solapan.

Capacidad de los programas

Mayor número de bots con, en promedio de la ventana de medición, menos de 50 ms de MSPT p95 en las tres repeticiones, todos con Aikar flags.



Capacidad de los programas con un mínimo de 19,5 TPS

Añadido a posteriori: mayor número de bots con, en promedio, al menos 19,5 TPS en las tres repeticiones.



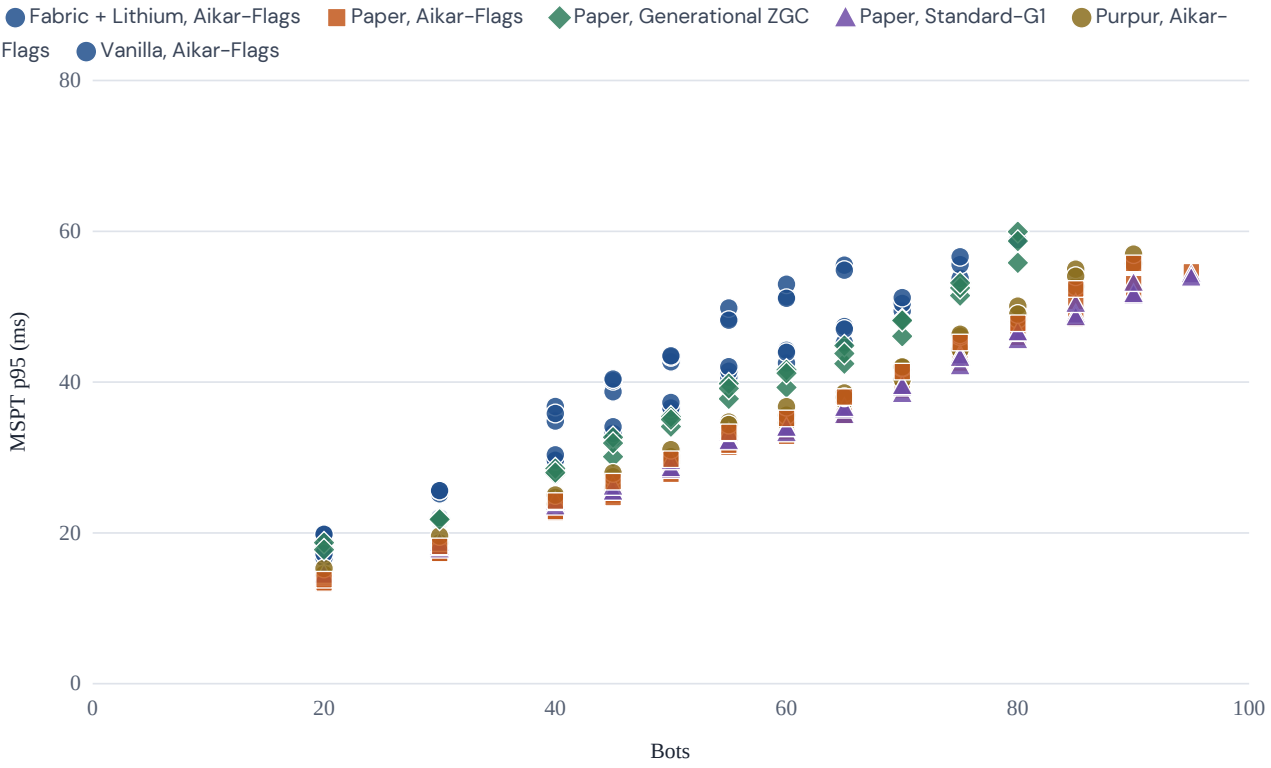
Capacidad de Paper por perfil de JVM

Mayor número de bots con, en promedio de la ventana de medición, menos de 50 ms de MSPT p95 en las tres repeticiones.



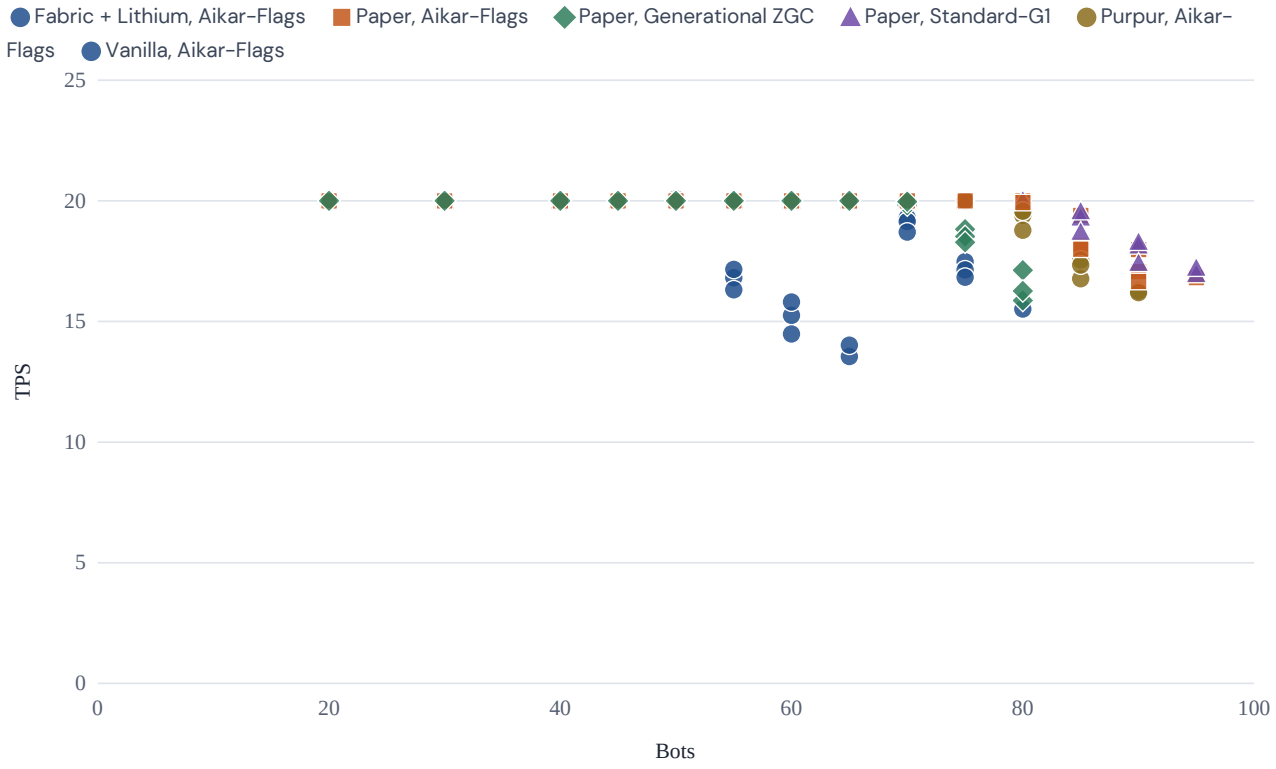
Tiempo de tick p95 por nivel

Cada punto es un nivel de una ejecución, representado como media de la ventana de medición. El límite de 50 ms se aplica a esta media.



Tasa de ticks real por nivel

Cada punto es un nivel de una ejecución. El objetivo son 20 ticks por segundo.



Todas las combinaciones

Combinación	Capacidad	Ejecución más alta	A 19,5 TPS	Ejecución más alta (TPS)	p95 a 50, mín	p95 a 50, máx	p99 a 50, mín	p99 a 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

Las 18 ejecuciones

N.º	Combinación	Rep.	Capacidad	TPS ahí	A 19,5 TPS	Válida
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

Comparaciones por pares

Una diferencia se considera demostrada si los rangos de las tres repeticiones no se solapan.

Magnitud	A	A mín	A máx	B	B mín	B máx	Demostrada
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

Magnitud	A	A mín	A máx	B	B mín	B máx	Demostrada
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 Método

1 **Fijar los programas**

Todos los programas se ejecutaron en Minecraft 26.3. Java, los programas de servidor y Lithium se comprobaron con la suma de verificación publicada. Para el iniciador de Fabric no hay una suma de verificación publicada; en ese caso se registró el valor SHA-256 propio. Los server.properties fueron iguales para todos los programas y los demás ajustes se dejaron en los valores predeterminados de cada programa. El único cambio adicional afecta a Paper y Purpur: se desactivó el limitador de conexiones por dirección IP, porque todos los bots proceden de la misma dirección de loopback. Lithium se describe a sí mismo como un mod de optimización que no debe alterar la mecánica del juego (fuente 4), y Purpur como un sustituto directo de Paper (fuente 5).

2 **Un mundo común**

El servidor Vanilla sin modificar generó una vez, con la semilla 20261121, un mundo con 31.329 chunks totalmente generados. Cada ejecución partió de una copia nueva de ese mundo.

3 **Núcleos separados**

El servidor se ejecutó fijado a los núcleos 0 a 3 y el generador de carga a los núcleos 4 a 6. El script de medición no estuvo vinculado a ningún núcleo. Durante la medición, los datos en bruto se comprimían cada 15 minutos con la prioridad más baja en el núcleo 7. La entrada messung: 7 de plan.json no se aplicó como vinculación a un núcleo. El servidor y RCON solo eran accesibles en la dirección de loopback; online-mode=false se estableció únicamente en este montaje aislado. Erik Friedl aceptó el EULA de Minecraft para estos servidores de prueba.

4 **Bots de protocolo**

Un cliente propio y ligero habla el protocolo de red real de la versión 26.3: inicio de sesión, configuración, fase de juego, keep-alive y confirmaciones de teletransporte. El cliente envía confirmaciones de chunk cuando detecta un mensaje de chunk batch. Esto ocurrió en Vanilla y Fabric; en Paper y Purpur no contó ningún mensaje de ese tipo en ninguna ejecución. La base de datos de protocolos que usa mineflayer aún no incluía la versión 26.3 en el momento de la medición. Cada bot tiene una semilla fija, envía con una cadencia objetivo de 20 paquetes de posición por segundo y se mueve a velocidad de paso (0,2158 bloques por tick) entre puntos de ruta aleatorios. Cada seis bots forman un grupo, y los grupos están separados 256 bloques entre sí. Los bots vuelan en modo creativo a una altura de 120.

5 **Niveles y ventana de medición**

Los niveles fueron 20, 30 y 40 bots, y después en pasos de cinco. Tras la incorporación siguieron 60 segundos de tiempo de estabilización y 120 segundos de ventana de medición con una muestra cada 5 segundos. Una ejecución terminaba tras dos niveles por encima del límite o por debajo de 15 TPS.

6 **Magnitudes medidas**

MSPT p50, p95 y p99 procedieron en todos los programas del comando /tick query por RCON, que evalúa los últimos 100 ticks. El valor del nivel es la media de estos bloques en la ventana de medición. Los TPS se calcularon a partir del avance del tiempo de juego. Además se registraron la RAM del proceso de Java, la CPU por grupo de núcleos, el CPU steal y el número de bots confirmado mediante la lista de jugadores.

7 **Pruebas previas al plan**

Antes de fijar el plan se hicieron dos ejecuciones piloto (Paper, Vanilla) y breves ejecuciones de prueba de todas las combinaciones. Después se fijaron la rejilla de niveles y los tiempos de medición; el límite de 50 ms ya estaba establecido antes. Las pruebas previas no se incluyen en la evaluación.

8 **Validez**

Fijado de antemano: CPU steal por debajo de 2 %, generador de carga por debajo de 70 % de CPU y por debajo de 20 ms de retraso de temporizador (p99), muestras completas, todos los bots conectados mientras el servidor no esté sobrecargado, y dos bots de muestra se mueven de forma demostrable en la ventana de medición. Las ejecuciones no válidas se habrían repetido; no hubo ninguna.

9 **Capacidad y diferencias**

Fijado de antemano: la capacidad de una ejecución es el nivel más alto hasta el cual todos los niveles se mantienen, en promedio de la ventana de medición, por debajo de 50 ms de MSPT p95 y todos los bots están conectados. La capacidad de una combinación es el menor valor de sus tres ejecuciones. Una diferencia solo cuenta si los rangos de las repeticiones no se solapan.

10 **Complemento posterior**

Tras la medición se vio que la tasa de ticks caía en parte por debajo de 20 TPS mientras el tiempo de tick p95 aún estaba, en promedio, por debajo de 50 ms. Por eso se evaluaron además la capacidad con un mínimo de 19,5 TPS y las magnitudes con 50 bots. El nivel 50 es el más alto en el que todas las combinaciones alcanzaron aún, en promedio de la ventana de medición, al menos 19,5 TPS. Estas evaluaciones no estaban fijadas de antemano y así se indica en los datos.

Configuración de la medición

Versión del juego	Minecraft 26.3 (protocolo 777)
Vanilla	26.3
Paper	26.3, compilación 257
Purpur	26.3, compilación 2660
Fabric	Loader 0.20.1, Installer 1.1.2
Lithium	0.26.4 para 26.3
Java	Temurin 25.0.5.1+2
Heap	10 GB (Xms igual a Xmx)
Perfil de JVM G1 estándar	-XX:+UseG1GC
Perfil de JVM Aikar flags	Flags según la documentación de PaperMC
Perfil de JVM Generational ZGC	-XX:+UseZGC
Máquina virtual	8 vCPU bajo KVM, 31,3 GiB de RAM, 100 GiB de disco virtual, Debian 12
CPU informada	QEMU Virtual CPU, AuthenticAMD; el modelo físico no se puede determinar desde la VM
Núcleos	Servidor 0 a 3, bots 4 a 6, copia de seguridad 7, script de medición sin vincular
Distancia de visión	10 chunks, distancia de simulación 10 chunks
Modo de juego	Creativo, dificultad normal
Periodo de medición	21.11.2026, 20:12 UTC hasta 22.11.2026, 06:59 UTC

5 Muestra y periodo

Periodo	21/11/2026 a 22/11/2026
Población del estudio	18 Ejecuciones de medición
Evaluado	18 Ejecuciones de medición
Selección	Seis combinaciones de programa y perfil de JVM con tres repeticiones cada una, en un orden aleatorio fijado de antemano.

6 Límites del alcance de los resultados

Los bots no son jugadores. Los bots vuelan, no construyen, no luchan, no chatean, no usan inventario y no activan circuitos de redstone. Los jugadores reales generan otra carga. Por eso las cifras absolutas de bots no se pueden trasladar a un servidor real. El mundo se generó de antemano y los bots no salieron de esa zona. Por tanto, la carga de la generación de nuevos chunks no está incluida.

Una VM. Se midió en una única máquina virtual con cuatro núcleos para el servidor. En otro hardware los valores serán distintos, y también las distancias entre los programas pueden cambiar.

Un perfil de prueba. Hubo un mundo, una semilla, un perfil de movimiento y, aparte de Lithium, ningún mod ni plugin. Los programas no se optimizaron de forma individual. Con una configuración adaptada u otra carga, el orden puede ser distinto. El ciclo de día y noche estuvo activo. Como todas las ejecuciones siguen el mismo calendario, determinados niveles caen siempre en la noche. En esta medición no se registró su influencia por separado.

Configuración predeterminada distinta. La configuración predeterminada de Paper y Purpur contiene el ajuste entity-activation-range (animales y monstruos 32 bloques), que falta en las plantillas de Vanilla y Fabric. No se comprobó si todos los programas calcularon el mismo trabajo de simulación, y no se registraron los números de entidades.

MSPT y TPS divergen. En Vanilla, el tiempo de tick p95 con 55 bots estuvo, en promedio de la ventana de medición, por debajo de 50 ms, pero la tasa de ticks fue de entre 16,31 y 17,16 TPS. No se investigó la causa. En Vanilla, la capacidad fijada de antemano queda por ello 5 bots por encima del valor con un mínimo de 19,5 TPS. En las otras cinco combinaciones ambos valores mínimos coinciden.

Medias en lugar de valores individuales. El límite de 50 ms se aplica a la media de los bloques de 100 ticks en la ventana de medición. En 13 de 18 ejecuciones, en el nivel de capacidad al menos un bloque individual estuvo en 50 ms o por encima, como máximo en 54,2 ms. También el límite de 19,5 TPS se aplica a la media de la ventana de medición. En los niveles de la capacidad con un

mínimo de 19,5 TPS, algunas muestras individuales de 5 segundos quedaron por debajo, la más baja con 17,62 TPS (Vanilla, 50 bots).

Deriva a lo largo de la noche de medición. Con el mismo número de bots, el tiempo de tick p95 aumentó con la posición en el orden de las ejecuciones, con 50 bots en torno a 0,11 ms por ejecución (evaluado a posteriori). En esta medición no se investigó la causa. El orden fue aleatorio, pero no equilibrado por bloques, por lo que la dispersión de las repeticiones incluye también este efecto.

Magnitudes posteriores. La capacidad con un mínimo de 19,5 TPS, el nivel de comparación con 50 bots y la comparación p99 entre Aikar flags y G1 estándar se eligieron solo después de ver los datos. Son observaciones complementarias y no comprobaciones fijadas de antemano.

Rejilla gruesa, tres repeticiones. Los niveles están separados cinco bots en el rango relevante; las diferencias menores no son medibles. Tres repeticiones por combinación muestran la dispersión solo de forma aproximada. No se calculó ninguna prueba estadística.

Valores de RAM no comparables. Las Aikar flags contienen AlwaysPreTouch, que reserva la memoria ya al arrancar (fuente 1). Por ello la memoria medida depende del ajuste de la JVM y no muestra la necesidad del programa. Este estudio no hace ninguna afirmación sobre qué programa necesita menos RAM.

Comparación de JVM solo con Paper. Los tres perfiles de JVM se midieron solo con Paper, un heap de 10 GB y Java 25. La comparación de programas se hizo solo con Aikar flags. Ambos resultados no se pueden trasladar sin más a otras combinaciones.

Instantánea. Los resultados son válidos para las versiones y compilaciones indicadas. Las versiones posteriores pueden comportarse de otra manera.

7 Datos y verificabilidad

Para repetirlo se necesita una máquina Linux con ocho núcleos y unos 32 GB de RAM como aquí, Node.js y Python 3. Los scripts esperan la carpeta de trabajo /root/studien/haertetest, descargan Java y las mismas versiones de los programas mientras esas descargas sigan disponibles, comprueban las sumas de verificación, generan el mundo y ejecutan el plan de medición. La semilla, el orden, los niveles y las reglas figuran en plan.json. Otro hardware, otras compilaciones u otra carga del sistema anfitrión producen otros valores absolutos.

Archivo	Tamaño	SHA-256
daten.csv Valores medidos por nivel (CSV)	35,6 KB	a2c2bfdfcc8ddfce2f6eac4044c15aeb ed57f4d0fab95b8a48af72c47e2f9e9e
daten.json Valores medidos por nivel (JSON)	94,6 KB	f798d400aba35617cf30f8c0cd2ad07c f6aace6e19e02300a0a1b822c6ac980e

Archivo	Tamaño	SHA-256
laeufe.csv Resultado por ejecución	2,7 KB	3a7f0af91987e982f8e95f97c60ef833 0ea9ba8a9f4688e76c64f8514b3486d1
zusammenfassung.csv Resultado por combinación	1,1 KB	8ba8269c39e20d767c2a8c87dfb805fd 0d937bb59bd448680f23401f0e66bd88
vergleiche.csv Comparaciones por pares	4,7 KB	99575f03f6a3f2075bcd20426a18e86d 6d14ad5410618ebb101050c3fabcda9d
p99-aikar-g1.csv p99 Aikar flags y G1 estándar	0,5 KB	4d0b6c017068a81642cddc339b1623f2 71d5d543abaf0ea9e4722152907e64ac
proben.csv Muestras individuales	517,9 KB	7fa08cf87e94c1e58c680ae9cec668ba 6d56da228bda6be08849d8232db27c33
plan.json Plan de medición	5,9 KB	cc335b804f52374f6c77afda5c55dd19 dbf5fa03d595c8f3a9bb72cedd569192
messprotokoll.txt Registro de medición	8,1 KB	f234fa1d0a16b312777dd0b93fb12f0e b13161fbdbb13125501e034837705a64
herkunft.txt Origen de los programas	1,4 KB	5f9e921b1c4fe697d2e676b7b5777eee ae784a4c4418b211230e7ac333bb676d
liesmich.txt Léame	1,4 KB	8a444fae54933207d6f896459ac23511 c541a656d0fc2da2a26b2f69091f0935
01-einrichten.sh Instalación	3,4 KB	27627241f4e7964be26e9b3ac2b3c1de 74482e323348569f6e7ba6e9290b7ea9
02-vorlagen.py Mundo y plantillas	4,1 KB	fc70d06d68b18d70425e0987794f06da 257659424a168433c2048577258d4cd5
03-plan.py Plan de medición	3,5 KB	773d40d0d4dd1323c748a8c383ecf1ff b0fb91299cbea545f21144dfa083a034
gemeinsam.py Partes comunes	5,3 KB	da16b51f5b78fd55c070c1eaa9b43485 cd7fd1654ce8c47e71df87b26bcb7e20
messlauf.py Ejecución de medición	10,9 KB	395ec326a334855d03a333879da16f90 0730d0307f06065f078f79af8de21637
hauptlauf.py Medición principal	4,6 KB	7b1602088aba30734e706e35ab894d4c 918a91eec8b0846a68a2ee392c6cec2f
bots.mjs Generador de carga	14,3 KB	6fa44c58b3c56a31fb77fb19602cc06a ab93f08e8e8984a40bb95f2ad1064a10
welt-pruefen.py Comprobación del mundo	1,5 KB	e0bb7dcc1e2327e9b916355b3e3dba55 2751a2f6c13989ffd07f10c234c0940e
auswertung.mjs Evaluación	10,0 KB	42dc3452a2061bad7ede2d84610cdf8c 8ef3adb5e7c6c7b8593c502aa2b85a33
baue-studie.mjs Construcción del estudio	39,2 KB	c0e76feab583afa11baf3bbce105203f 4a3ebcf57e66adf44380bb7854a8b90b

Campos de los archivos de datos

Campo	Unidad	Significado
lauf		Identificador de la ejecución de medición con la posición en el orden aleatorio
kombination		Identificador formado por programa y perfil de JVM
anzeige		Nombre legible de la combinación
programm		Programa de servidor
jvm		Perfil de JVM
wiederholung		Número de la repetición, de 1 a 3
versuch		Intento dentro de la repetición, 1 es el primero
reihenfolge		Posición en el orden aleatorio fijado
gueltig		La ejecución cumple todas las reglas de validez fijadas de antemano
lauf_gueltig		La ejecución correspondiente es válida
spieler	Bots	Número objetivo de bots en este nivel
spieler_min	Bots	Menor número de bots confirmado mediante la lista de jugadores en la ventana de medición
spieler_verbunden	Bots	Número de bots confirmado mediante la lista de jugadores en esta muestra
unter_grenze		MSPT p95 en promedio de la ventana de medición por debajo de 50 ms y todos los bots conectados
p50_ms	ms	Mediana del tiempo de tick; en los niveles, media de los bloques de 100 ticks
p95_ms	ms	Percentil 95 del tiempo de tick; en los niveles, media de los bloques de 100 ticks
p95_max_ms	ms	Mayor valor p95 de un bloque de 100 ticks en la ventana de medición
p99_ms	ms	Percentil 99 del tiempo de tick; en los niveles, media de los bloques de 100 ticks
mspt_mittel_ms	ms	Tiempo de tick medio según /tick query
tps	Ticks/s	Tasa de ticks real a partir del avance del tiempo de juego; en los niveles, media de la ventana de medición
tps_min	Ticks/s	Menor tasa de ticks de una muestra de 5 segundos en la ventana de medición

Campo	Unidad	Significado
ram_gb	GiB	Memoria ocupada del proceso de Java (RSS)
cpu_server_pct	%	Utilización media de los cuatro núcleos del servidor
cpu_bots_pct	%	Utilización media de los tres núcleos del generador de carga
steal_pct	%	CPU steal en los ocho núcleos
bots_verzug_p99_ms	ms	Mayor retraso p99 de los temporizadores en el generador de carga
getrennt	Bots	Bots desconectados de forma inesperada hasta este nivel
proben		Número de muestras de 5 segundos en la ventana de medición
zeit_unix	s	Momento de la muestra como hora Unix
erste_ueberlast	Bots	Primer nivel cuyo MSPT p95 medio queda por encima del límite de 50 ms
kapazitaet	Bots	Capacidad fijada de antemano; en el resumen, menor valor de las tres ejecuciones
tps_bei_kapazitaet	Ticks/s	Tasa de ticks media en el nivel de la capacidad
kapazitaet_tps	Bots	Nivel más alto hasta el cual todos los niveles mantuvieron, en promedio de la ventana de medición, al menos 19,5 TPS; en el resumen, menor valor
kapazitaet_tps_190	Bots	Como kapazitaet_tps, con 19,0 TPS como límite
kapazitaet_tps_198	Bots	Como kapazitaet_tps, con 19,8 TPS como límite
p95_bei_50_ms	ms	MSPT p95 en el nivel con 50 bots
p99_bei_50_ms	ms	MSPT p99 en el nivel con 50 bots
tps_bei_50	Ticks/s	Tasa de ticks en el nivel con 50 bots
stufen_gemessen		Número de niveles medidos de la ejecución
ungueltig_grund		Motivo en las ejecuciones no válidas, en caso contrario vacío
im_programmvergleich		La combinación pertenece a la comparación de programas
im_jvm_vergleich		La combinación pertenece a la comparación de perfiles de JVM
laeufe_gueltig		Ejecuciones válidas de la combinación

Campo	Unidad	Significado
laeufe_ungueltig		Ejecuciones no válidas de la combinación
kapazitaet_median	Bots	Mediana de la capacidad entre las ejecuciones
kapazitaet_max	Bots	Mayor valor de capacidad de las ejecuciones
kapazitaet_tps_median	Bots	Mediana de kapazitaet_tps
kapazitaet_tps_max	Bots	Mayor valor de kapazitaet_tps
tps_bei_kapazitaet_min	Ticks/s	Menor tasa de ticks en la capacidad entre las ejecuciones
p95_bei_50_min_ms	ms	Menor valor p95 con 50 bots
p95_bei_50_max_ms	ms	Mayor valor p95 con 50 bots
p99_bei_50_min_ms	ms	Menor valor p99 con 50 bots
p99_bei_50_max_ms	ms	Mayor valor p99 con 50 bots
ram_bei_50_gb	GiB	RSS medio con 50 bots
cpu_bei_50_pct	%	Utilización media de los núcleos del servidor con 50 bots
vergleich		Comparación de programas o comparación de JVM
kennzahl		Magnitud comparada
regel		Si la magnitud se fijó de antemano o se añadió a posteriori
a		Primera combinación
a_min		Menor valor de la primera combinación
a_max		Mayor valor de la primera combinación
b		Segunda combinación
b_min		Menor valor de la segunda combinación
b_max		Mayor valor de la segunda combinación
unterschied_belegt		Los rangos de las repeticiones no se solapan
besser		Combinación con el valor más favorable, solo en caso de diferencia demostrada
aikar_min	ms	Menor valor p99 con Aikar flags en el nivel
aikar_max	ms	Mayor valor p99 con Aikar flags en el nivel
g1_min	ms	Menor valor p99 con G1 estándar en el nivel
g1_max	ms	Mayor valor p99 con G1 estándar en el nivel
aikar_niedriger		Los tres valores de Aikar quedan por debajo de los tres valores de G1

Campo	Unidad	Significado
g1_niedriger		Los tres valores de G1 quedan por debajo de los tres valores de Aikar

8 Fuentes

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

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

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

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

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

9 Transparencia

Interés propio

Nexthosting vende servidores de juegos, entre ellos servidores para Minecraft, y por ello tiene un interés económico en este tema. Ninguno de los programas probados procede de Nexthosting. La medición se realizó en una VM de prueba contratada por Nexthosting. El estudio no valora ofertas de proveedores de hosting. La responsabilidad de este estudio recae en Nexthosting, contacto: support@nexthosting.net.

Autoría

Erik Friedl, Nexthosting. El editor es Nexthosting.

Licencia

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

10 Cómo citar este estudio

APA

Friedl, E. & Nexthosting. (2026). Prueba de resistencia del servidor: ¿cuántos bots de protocolo soportan Vanilla, Paper, Purpur y Fabric con Lithium? (Versión 1.0.0) [Informe]. Nexthosting. https://nexthosting.net/es/estudios/server-haertetest-minecraft

BibTeX

```
@techreport{nexthosting2026serverhaertetestminecraft,
  author     = {Erik Friedl and {Nexthosting}},
  title      = {Prueba de resistencia del servidor: ¿cuántos bots de protocolo soportan Vanilla, Paper,
Purpur y Fabric con Lithium?},
  institution= {Nexthosting},
  type       = {Estudio},
  year       = {2026},
  version    = {1.0.0},
  url        = {https://nexthosting.net/es/estudios/server-haertetest-minecraft},
  urldate    = {2026-11-22},
  language   = {spanish}
}
```

11 Cambios y correcciones

Versión	Fecha	Tipo	Cambio
1.0.0	22/11/2026	Versión	Primera versión.