

TSR

Comprimere di più dove gli altri si fermano, in un tempo che si sceglie, con un archivio che è una funzione del dato.

White paper

Il metodo TSR: tre idee, le misure che le sostengono e i limiti dichiarati. Per chi valuta la tecnologia prima del prodotto.

- **Il tempo come unità della compressione.** Si sceglie quanto tempo dedicare; TSR calibra la macchina e i file reali e dichiara prima di partire durata, spazio e tempo di estrazione.
- **Lo strato incompressibile si smonta.** PDF, Office, PNG, JPEG e video si ricodificano dall'interno e tornano identici al byte: primo su dieci strumenti sui documenti d'ufficio.
- **L'archivio è una funzione del dato.** Stessi byte su tre architetture, formato congelato con specifica e vettori, integrità per blocco e per file, uso senza estrarre.

DOCUMENTO

TSR-WHP-06

EDIZIONE

27 settembre 2026

DIFFUSIONE

Pubblica

TITOLARE

Matteo Luigi Feroldi

DOCUMENTO PUBBLICO

Descrive il metodo e i risultati; il funzionamento interno del codec resta nel dossier tecnico, consegnato con un accordo di riservatezza. Ogni cifra porta corpus, macchina, profilo e data.

IN SINTESI

La compressione forte esisteva già. Quello che mancava era renderla abbastanza veloce da usarla e abbastanza onesta da fidarsene.

TSR (Time Space Reducer) è un compressore e archiviatore per macOS, Windows e Linux, con un motore scritto da zero in Rust. Queste pagine espongono il metodo che lo distingue, le misure che lo sostengono e i limiti dichiarati, per chi valuta la tecnologia: architetti, responsabili dell'infrastruttura, revisori tecnici.

47,888%

Documenti d'ufficio
1.322 file reali da fonti pubbliche: primo su dieci strumenti. Il migliore fra i rivali: 51,98 %

21,975%

enwik8
Cento megabyte di testo, in 19 secondi su tutti i core. 7-Zip al massimo: 24,80 % in 41

-18,2%

Fotografie JPEG
Ricodificate dall'interno e restituite identiche al bit, dove gli archiviatori generici restano a zero

3

Architetture
Apple Silicon, x86-64 e ARM64 di classe server producono gli stessi byte, verificato con le impronte

Prima idea**Il tempo come unità**

Si sceglie quanto tempo dedicare; TSR calibra macchina e file reali, prende il codec più forte che rientra e dichiara prima durata, spazio e tempo di estrazione. La famiglia forte gira su blocchi indipendenti, uno per core.

Seconda idea**Lo strato incompressibile si smonta**

I file già compressi (PDF, Office, PNG, JPEG, video, gzip) si smontano fino allo strato che li rende tali; i dati veri passano al codec e lo strato torna al suo posto byte per byte, verificato su ogni file.

Terza idea**L'archivio è una funzione del dato**

Lo stesso input dà gli stessi byte su tre architetture; ogni blocco e ogni file portano l'impronta; il formato è congelato con specifica e vettori; l'archivio si usa mentre resta compresso.

Indice

01	Il problema	3	06	Le prove e il metodo	11
02	Il tempo come unità della compressione	4	07	Perimetro dichiarato	13
03	Lo strato incompressibile	6	08	Verificare e adottare	14
04	L'archivio come funzione del dato	8	—	Glossario e riferimenti	15
05	Il dato resta utilizzabile	10			

01 Il margine dei compressori classici è esaurito

E la parte del dato che cresce più in fretta è già compressa: per un archiviatore tradizionale è indistinguibile dal rumore.

Lo spazio è la voce di costo che cresce da sola

Ogni copia di un dato paga il suo rapporto di compressione: l'archivio primario, il backup, la replica fuori sede, il nastro, ogni trasferimento in rete. Un punto di rapporto guadagnato una volta agisce su tutte le copie, per tutta la vita del dato. La capacità si compra a lotti, le finestre di backup restano quelle che sono e la banda si paga al byte: il rapporto è la sola leva che si paga una volta e rende ogni notte.

Due famiglie, e un limite pratico

Gli archiviatori di uso comune (zip, 7-Zip, xz, zstd, brotli) appartengono alla famiglia nata col DEFLATE del 1993: cercano ripetizioni e codificano ciò che resta. Da anni rifiniscono le stesse idee e si muovono di frazioni di punto: su enwik8, cento megabyte di testo, stanno fra il 24,8 e il 25,7 %. I compressori che li superano (PAQ, zpaq, cmix) appartengono alla famiglia dei contesti mescolati, che predice ogni bit da decine di indizi e li pesa insieme: vale da tre a otto punti in più e costa da dieci a mille volte il tempo su un solo core. Per vent'anni è rimasta uno strumento da laboratorio.

La parte che cresce è già compressa

PDF, documenti Office, PNG, JPEG, video, archivi gzip: i formati con cui il dato viene prodotto contengono già uno strato di compressione. Un archiviatore generico li tratta come rumore e restituisce dal 96 al 99,5 % dell'originale: il suo margine è esaurito. Quello che resta sta dentro quei formati, negli strati di codifica del 1992 e del 1993, e ci si arriva soltanto rimettendo tutto al suo posto, byte per byte.

CHE COSA RESTITUISCONO GLI ARCHIVIATORI DOVE IL DATO È GIÀ COMPRESSO

INSIEME DI FILE	7-ZIP AL MASSIMO	UN ALTRO RIVALE
Fotografie JPEG (galleria, 41,6 MB)	≈ 100 %	generici ≈ 100 %
Foto e video dal telefono (scenario «famiglia»)	97,41 %	zip 98,02 %
Film e master (scenario «videomaker»)	98,68 %	zip 99,45 %
Documenti d'ufficio (1.322 file, 199,5 MB)	52,47 %	zpaq 51,98 %

Tre costi che il rapporto governa

Lo spazio occupato su ogni livello di storage; la durata delle finestre di backup e di replica; i byte trasferiti in rete, che si pagano a consumo.

La scala di enwik8

bzip2 29,01 % · zstd 25,27 % · 7-Zip 24,80 % · PPMd 21,41 % · **TSR 21,975 %** · bsc 20,80 % · zpaq 19,63 %. Tutti sulla stessa macchina, ai livelli massimi documentati; la pagina 5 mette accanto i tempi.

Da dove nasce questo documento

Da un progetto che ha scritto ogni cifra con la sua misura accanto e ha pubblicato i verdetti negativi come i positivi. Le pagine 11–13 dicono metodo, corpora e limiti.

02 Il tempo come unità della compressione

L'utente sceglie quanto tempo dedicare; TSR converte quel tempo in rapporto sulla sua macchina e sui suoi file, e dichiara il risultato prima di partire.

I livelli da 1 a 9 degli archiviatori descrivono l'algoritmo: lo stesso livello dura un minuto su un computer e dieci su un altro, e lo spazio restituito si scopre alla fine. Il tempo è la sola grandezza che chi comprime sa negoziare, ed è quella naturale di un codec che converte tempo in rapporto.



Il flusso di una compressione a tempo: calibrazione e piano si ripetono su ogni lotto, e a lavoro finito la stima si confronta con il risultato.

Calibrare sulla macchina vera, sui file veri

Prima di comprimere, TSR campiona i file reali del lotto: strisce per fasce di dimensione, pesate per la massa di byte che ogni fascia porta, e classi per tipologia (testo, binario, media già compresso). Su quei campioni misura, per ogni codec disponibile, velocità e rapporto sulla macchina che sta lavorando in quel momento. Poi sonda i filtri sui file veri, contando quanto rendono e quanto costano, e stima i duplicati. Il tutto costa da uno a tre secondi.

Il piano

Il budget di tempo si confronta con le stime e vince il codec più forte che rientra. Cinque livelli (Velocissimo, Veloce, Medio, Lento, Molto lento) sono multipli del tempo minimo della macchina ($\times 1,1$, $\times 3$, $\times 8$, $\times 20$, $\times 60$); un budget scritto in secondi si prende alla lettera. Prima di partire TSR mostra durata, spazio risparmiato e tempo di estrazione stimati; a lavoro finito confronta la stima con il risultato.

Le stime sono prudenti

La direzione dell'errore è dichiarata: il rapporto è misurato su strisce e i blocchi interi scaldano il modello una volta sola, quindi il risultato reale è quasi sempre migliore dell'annuncio. Sui due archivi ZIP di un utente il profilo bilanciato ha stimato 12 secondi per 13 reali e 5,3 MB di risparmio per 5,27; sui dati veri di un backup di posta il livello Medio ha dichiarato 21 secondi per 20. La stima del tempo di estrazione sta fra -2% e $+16\%$.

Quattro profili a codec fisso

Per gli script e i servizi esistono anche i profili senza calibrazione: veloce, bilanciato, backup, massimo. Un servizio evita due secondi e mezzo di misura a ogni lavoro, e con un codec fisso lo stesso input dà lo stesso archivio su ogni macchina.

Quando il contenuto è già compresso

La calibrazione lo dice subito, prima di partire: la riga del piano riporta che cosa hanno misurato le sonde («pdf 1,9 MB/s · flusso $\times 16$ · -15% », «contenitori: non si applica») e con quale livello conviene procedere.

Un budget fuori portata

Un tempo più corto del minimo raggiungibile viene segnalato insieme al minimo, e a quello che si otterrebbe con esso.

Il codec non fa sconti a nessuno

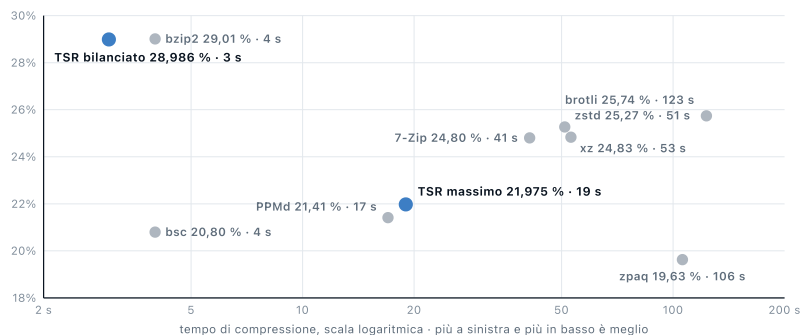
La calibrazione cronometra i codec interfogliati per striscia: una deriva del processore durante la misura colpisce tutti allo stesso modo, e il rapporto fra le velocità, su cui decidono i livelli, resta stabile. Verificato il 22 settembre 2026 con otto corse vere su otto allo stesso piano.

Blocchi indipendenti, un blocco per core

La modellazione a contesti mescolati è seriale per natura: ogni bit dipende dalle tabelle aggiornate dal bit precedente. TSR la rende usabile sulle macchine di oggi dividendo l'input in blocchi indipendenti (da 2 a 16 MiB secondo il profilo, sempre dentro un file) e codificandone uno per core. I modelli ripartono da zero a ogni blocco: costa fra l'1 e il 2 % di rapporto e compra tre cose insieme: il parallelismo in scrittura e in lettura, l'accesso casuale a qualunque punto dell'archivio, e la localizzazione di un danno al singolo blocco. Su Silesia la pipeline a blocchi comprime 3,4 volte più in fretta dei lotti per file, estrae 5,5 volte e verifica 5,7.

Il risultato sul corpus di riferimento: enwik8 al 21,975 % in 19 secondi su tutti i core di un Apple M5 Pro, dove zpaq fa 19,63 % in 106 secondi su un core solo e 7-Zip al livello massimo 24,80 % in 41.

enwik8: rapporto contro tempo di compressione, Apple M5 Pro



Rapporto in percentuale dell'originale contro tempo di compressione; rivali ai livelli massimi documentati, misure dell'8 settembre 2026. **I tempi sono a corsa singola** e danno l'ordine di grandezza; le dimensioni sono deterministiche.

Tre corsie per tre esigenze

Il codec ha tre corsie, e il profilo o il piano scelgono. La corsia veloce lavora a byte con una grammatica di ripetizioni e una codifica entropica a tabelle: sul testo 35,9 % a 119 MiB/s per core in compressione e 617 in decodifica, lo stesso rapporto di zstd al livello 3 a circa metà della sua velocità. La corsia intermedia cerca le ripetizioni con un parsing ottimale e codifica ciò che resta con i contesti: sui documenti d'ufficio 50,925 % in 23 secondi, dove zpaq impiega 140 per il 51,98 %. La corsia forte è il context mixing con l'elezione: sui budget ampi il codec prova più candidati sullo stesso blocco e tiene il più corto.

L'acceleratore grafico

La ricerca delle ripetizioni può girare sulla GPU (Metal, Vulkan) sotto un contratto: ogni risultato si verifica sulla CPU prima dell'uso, e i byte dell'archivio sono identici con e senza. Sul profilo massimo il tempo è uguale, acceso o spento, perché il collo di bottiglia è il context mixing; sul bilanciato va da 1,4 a 2,9 volte più veloce sugli scenari d'uso. Il comando `tsr gpu-check` dimostra il contratto sulla macchina di chi valuta.

La memoria

Il budget di memoria decide quanti blocchi girano insieme, cioè la velocità, mai i byte: lo stesso input dà lo stesso archivio a ogni tetto. Senza indicazioni il budget è una stima dai core (fra 1 e 8 GB); chi lo chiede con un tetto esplicito ottiene un contratto, misurato a 1,79 GB con un tetto di 2 su macOS, pagando fino a tre volte il tempo sugli alberi dove i filtri lavorano di più.

Il costo dei blocchi, misurato e accettato

La modalità solida e i blocchi da 64 MiB sono stati provati: 0,335 punti di rapporto, al prezzo dell'accesso casuale. Scartati, e il verdetto è scritto.

03 Smontare lo strato incompressibile, e rimetterlo identico al byte

Per un compressore generico un file già compresso è a otto bit per byte. TSR lo smonta fino allo strato che lo rende tale, ricodifica i dati veri e dimostra di saperlo rimontare prima di scrivere.

Involucro e tratti compressi

Un file compresso si scompone in un involucro (intestazioni, indici, le parti che TSR non sa riprodurre) e in tratti compressi. I tratti si aprono e i dati grezzi passano al codec del profilo; per ogni tratto si annota la combinazione esatta di parametri che lo riproduce. In estrazione l'involucro torna com'era e i tratti si ricomprimono con la stessa libreria, incorporata nel programma e identica su ogni sistema. È la strada di gzip, dei contenitori ZIP e Office, dei PNG e degli stream dei PDF.

Per il JPEG lo strato da smontare è la codifica Huffman del 1992: TSR la decodifica fino ai coefficienti della trasformata, mai fino ai pixel, e li ricodifica con un modello adattivo che predice ogni coefficiente dai blocchi vicini. Per i video H.264 lo strato è la codifica aritmetica, ricodificata gruppo di immagini per gruppo. In ogni caso il file estratto è identico al bit a quello di partenza.

Le tre guardie

- **Ogni file si ricostruisce prima di scrivere.** Il filtro si applica, il file si rimonta e si confronta byte per byte con l'originale; al primo byte diverso entra com'è.
- **Le due strade si confrontano.** Il flusso smontato e il file com'è si comprimono tutti e due col codec del profilo, e vince il più corto, file per file. Senza questo confronto un archivio poteva arrivare al 200 % dell'originale: trovato dalla misura il 6 agosto 2026, corretto lo stesso giorno.
- **L'incompressibile si riconosce prima di lavorare.** Una guardia campiona testa, quarti e coda di ogni blocco e risponde «com'è» prima di eseguire i codec: sui dati casuali il profilo massimo risparmia il 94 % del tempo.

Un filtro può quindi soltanto far scendere la dimensione dell'archivio.



Il cammino di un file già compresso; le due uscite a tratteggio rendono sicura l'operazione.

Che cosa resta fuori, misurato

I file scritti con varianti proprie dell'algoritmo (alcuni PDF, gli stream cifrati, le immagini ottimizzate con zopfli o oxipng), i formati senza uno strato da smontare (MP3, AAC, JPEG 2000, WebP) e tutto ciò che è davvero casuale o cifrato entrano tali e quali. La copertura è per singolo flusso: il resto del lotto guadagna comunque.

La libreria che ricomprime

La zlib che riproduce i tratti compressi è incorporata nel programma e fissata dal file delle dipendenze. Un aggiornamento che cambiasse un byte dell'uscita renderebbe irricostruibili gli archivi: i test che congelano i flussi lo bloccano.

In estrazione

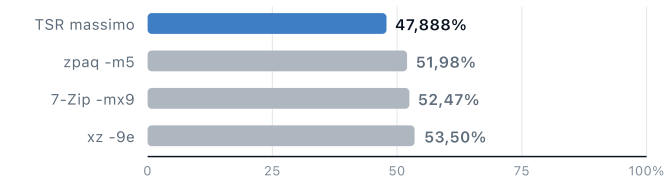
I file smontati si rimontano su una squadra di ricostruttori, più file insieme, dentro il budget di memoria: i gzip ricompresi si estraggono a 178 MB/s aggregati contro i 23 di un core solo (misura del 22 settembre 2026), identici alla sorgente.

Dieci famiglie di file, ognuna con la sua misura

FAMIGLIA	CHE COSA SI SMONTA	RESA MISURATA
JPEG	Lo strato Huffman, fino ai coefficienti della trasformata	−18,2 % sulle foto standard, −11,8 % sulle progressive; uno specialista ≈ −22 %
gzip, zlib	Il flusso DEFLATE, coi parametri ritrovati e verificati	−32 % sulla massa dei gzip riproducibili
ZIP, Office, JAR	Ogni membro DEFLATE, involucro intatto	−43,5 % sui contenitori Office moderni e sui JAR
PNG	Lo stream zlib delle immagini classiche	−22,9 % sui PNG classici; gli ottimizzati viaggiano com'erano
PDF	I soli stream DEFLATE che si riproducono	Dall'87–99 % al 50–79 % con la zlib standard dei produttori
Video H.264 (MP4, MOV, MKV)	La codifica aritmetica, per gruppo di immagini	2–3,7 % su film e riprese, dove i rivali valgono zero; una guardia salta i video che renderebbero poco
Eseguibili (x86, ARM64)	Un filtro sugli indirizzi dei salti	Famiglia eseguibile 26,66 % contro 28,22 % di xz
Audio PCM (WAV, AIFF, CAF)	Predizione lineare per blocco, canali separati	Corpus audio 59,444 % col massimo; sul PCM puro sei punti davanti a FLAC −8 (11,73 % contro 17,85 %)
Contenitori con DEFLATE irriproducibile	La migliore fra tre strade per membro, decisa su un campione	−18,5 % sui contenitori prima lasciati intatti
Immagini grezze a 16 bit	Un predittore sui pixel, residuo per piani di byte	Tomografie −3,63 %, risonanze −1,33 %; rifiuta dove peggiorerebbe

Il caso d'uso del prodotto: i documenti d'ufficio

199,5 MB in 1.322 file da fonti pubbliche (Office moderno e legacy, PDF di più produttori, EPUB, JAR), col 34,9 % della massa dentro contenitori DEFLATE. I dieci strumenti si fermano fra il 51,98 e il 62,40 %; TSR scende al 47,888 %, e il profilo bilanciato fa 50,925 % in 23 secondi dove zpaq impiega 140 per il 51,98.



Corpus costruito da fonti pubbliche (ECMA, Apache POI, Maven, Gutenberg, RFC Editor) con criteri eseguiti da uno script: chiunque lo ricostruisce e ripete la misura. Dieci strumenti misurati, in grafico i tre di riferimento; Apple M5 Pro, 8 settembre 2026.

Dove i contenitori mancano

Sui documenti pubblici di Govdocs1 (980 file, l'1,1 % della massa in contenitori) TSR è secondo: 48,598 % contro il 47,51 % di zpaq. Il vantaggio segue la quota di contenitori nel volume, ed è dichiarato così.

Film e riprese

Sui film H.264 il 2–3,7 % è misurato su riprese personali, quindi dichiarato senza piazzamento. Su una videoteca TSR vale per l'accesso immediato e per l'integrità.

Nel profilo veloce

Il profilo veloce tiene i due filtri leggeri (eseguibili e audio) e lascia le ricompessioni ai profili più lenti: col codec veloce rendono da 0,3 a 1,4 punti e costano il tempo di rimontarle. Deciso da una misura del 21 settembre 2026.

04 L'archivio è una funzione del dato

Lo stesso input, con lo stesso profilo, produce gli stessi byte su architetture e sistemi operativi diversi. Ogni blocco e ogni file portano la propria impronta. Il formato è congelato, con una specifica che si esegue.

Determinismo su tre architetture

PROVA	ESITO
Archivi interi, macOS e Linux, a parità di permessi e date	Impronte SHA-256 identiche 4 su 4 (enwik8 e Silesia, profili massimo e bilanciato), anche su 11.356 PNG
Windows	Contenuto compresso identico al byte su otto archivi; l'indice riflette i permessi che quel sistema dichiara sui file
Impronte per core	Identiche su Apple Silicon, x86-64 e ARM64 di classe server; con le istruzioni vettoriali accese e spente, 24 su 24
Scenari d'uso	Contenuto identico sulle tre architetture su otto persone su nove; la nona differisce solo su Windows, per 575 nomi con «?» riscritti dal filesystem
Acceleratore grafico	Archivi identici con la corsia accesa e spenta, su Metal e su Vulkan

La proprietà è possibile perché il codec lavora soltanto con aritmetica intera, i suoi flussi sono congelati da test che confrontano byte per byte, e ogni evoluzione futura prende un identificativo nuovo accanto ai vecchi. È la formulazione precisa, ed è quella che regge a una verifica: chiunque la ripete con un comando sulla propria macchina.

Integrità per blocco e per file

Ogni blocco porta l'impronta BLAKE3 del suo contenuto originale, e ogni file la sua: la verifica dice quale file è danneggiato e in quale blocco; l'indice ha la propria impronta. La verifica profonda ricostruisce anche i file smontati e li confronta con l'impronta per file. La stessa verifica esiste in una forma che va piano, si ferma e riprende da dove era (tsr scrub, con tetti di banda e di CPU misurati da fuori) e si programma una volta alla settimana.

A che serve

Conservazione a lungo termine, catena di custodia, verifica fra siti diversi, deduplicazione a monte, build riproducibili: tutte richieste che pretendono un archivio che sia funzione pura del dato.

Il formato 1.0

Congelato nel settembre 2026. Ogni archivio scritto da qualunque versione si legge con tutte le successive, per sempre: per ogni versione del formato esiste un archivio conservato come vettore, che la suite apre a ogni esecuzione e che si conserva com'è. La specifica, in inglese, è verificata da un lettore scritto soltanto da essa; i vettori sono quarantanove (quattordici storici, cinque del formato 1.0, trenta negativi) e li leggono tre lettori indipendenti.

Il rifiuto detto

Un lettore che incontra ciò che non conosce lo dice, nominando la cosa e la versione che legge («serve una versione più recente»): un errore pulito al posto di byte sbagliati. Ogni condizione di rifiuto della specifica ha il suo vettore negativo.

Cifratura di serie

Un archivio protetto da password è cifrato per intero con XChaCha20-Poly1305, indice compreso: nomi dei file, dimensioni, date e permessi restano segreti. La chiave si deriva dalla password con Argon2id, coi parametri scritti nell'archivio, così si possono alzare domani senza rompere ieri. Il progetto dichiara di aver scritto zero crittografia propria: primitive pubbliche e collaudate, usate come previsto. La cifratura è per blocco, quindi accesso casuale, streaming e volume montato funzionano identici sugli archivi cifrati; il costo misurato nel caso peggiore è di due centesimi di secondo e 4,2 KB su mezzo gigabyte. Una password sbagliata e un archivio manomesso producono due errori distinti.

Riparazione senza password

A richiesta l'archivio porta dall'1 al 30 % di parità Reed–Solomon: un danno che rientra nella quota, sparso o contiguo, si ripara per intero e l'archivio torna identico al byte, in un file nuovo, lasciando intatto l'originale. La parità copre i byte fisici, quindi un archivio cifrato si ripara senza conoscere la password: chi custodisce il supporto può ripristinarlo anche senza il diritto di leggerlo. Misura di riferimento: 34 MB col 10 % di parità, 100 KiB contigui più venti byte sparsi di danno, 46 frammenti su 8.732 ricostruiti in un decimo di secondo.

Durabilità alla caduta di corrente

Estendere un archivio scrive i blocchi nuovi dopo il vecchio indice, poi un indice e un piede nuovi in coda, con sincronizzazione su disco; il vecchio piede resta come ancora finché il nuovo è al sicuro. Il programma è stato interrotto di colpo 2.752 volte a istanti casuali durante creazione, estensione e compattazione, più il troncamento byte per byte: zero archivi persi, zero arresti anomali. Su errore o annullamento il disco resta com'era prima del lavoro.

La regola che tiene insieme le tre idee

Ogni byte dell'archivio è deducibile dal dato, dal profilo e dal formato pubblicato; ogni byte è verificabile con la sua impronta; ogni byte ignoto è rifiutato dicendolo. Da queste tre proprietà discendono la lettura fra macchine diverse, la verifica anche anni dopo e la riparazione senza fiducia in nessuno.

Aprire file di provenienza ignota

Il programma è scritto in Rust, che elimina per costruzione le classi di difetti più sfruttate nei parser. Ogni campo di dimensione ha un tetto: un archivio ostile di 299 byte con la dimensione gonfiata è respinto in un decimo di secondo. Il fuzzing a copertura gira su **18 bersagli** sotto rilevatore di errori di memoria: 11,77 milioni di esecuzioni registrate, zero difetti; a ogni push un controllo locale bloccante lo ripete per dieci secondi a bersaglio.

Rete e telemetria

La versione per le aziende lavora in locale e risponde soltanto a chi la interroga. La beta pubblica è una build a parte: controlla gli aggiornamenti, firmati, e manda crash e metriche solo col consenso. Il servizio di streaming ascolta di serie sulla sola interfaccia locale; esporlo a una rete richiede un segreto d'accesso.

Che cosa si vede dall'esterno

Di un archivio cifrato restano visibili la dimensione totale, quella dell'indice e quante volte è stato esteso. Il rollback dell'intero file resta possibile a chi possiede il supporto.

05 Il dato resta utilizzabile mentre è compresso

I blocchi indipendenti che rendono usabile la famiglia forte sono gli stessi che aprono l'archivio a metà: si decompime soltanto ciò che si tocca.

Accesso casuale per costruzione

Una lettura a un punto qualunque decompime i soli blocchi coinvolti, con una cache dimensionata sul budget. Un film H.264 da 2,3 GB dentro un archivio si riproduce e si scorre con 252 MiB di memoria e un salto di 1,6 secondi; in VLC e in QuickTime il fotogramma corrisponde al timecode a ogni salto, e il file resta compresso per tutto il tempo.

Tre porte sullo stesso archivio

- **Un volume di rete locale.** `tsr` serve espone l'archivio via HTTP con richieste parziali e come volume WebDAV di sola lettura: si monta nel Finder o in Esplora risorse senza estensioni del kernel, e ogni applicazione apre i file normalmente.
- **Una cartella, su Linux.** `tsr mount` monta l'archivio nel filesystem: `grep`, `ffmpeg`, un editor o una build aprono i file senza sapere che sono compressi. Misurato: 189 microsecondi per file, contro 1.198 del cammino HTTP e 18 dei file estratti.
- **L'interfaccia grafica.** Anteprima di testi e immagini, albero con ricerca, «dove va lo spazio», estrazione selettiva, verifica e riparazione, senza estrarre l'archivio.

Il secondo backup costa quello che è cambiato

Un file identico a uno già presente (nel lotto, o nell'archivio che si estende anche mesi dopo) diventa un riferimento; un blocco identico a uno già scritto si richiama invece di risciversi; col profilo di backup i blocchi si tagliano dove dice il contenuto, così un'inserzione riallinea entro un megabyte. Le tre cose vivono nello stesso contenitore.

Il prezzo dichiarato dell'accesso casuale

TSR comprime dentro i blocchi di un file e mai attraverso i file: sugli alberi di sorgenti ed e-mail 7-Zip in modalità solida sta davanti di 2-7 punti. È il prezzo dell'accesso diretto ai file, ed è scritto nel perimetro.

Un servizio che si può esporre

Fuori dall'interfaccia locale il segreto d'accesso è obbligatorio e si confronta a tempo costante; tetti di connessioni e di banda; metodi di scrittura respinti; la cifratura del trasporto a un reverse proxy davanti, con configurazioni pronte. Un demone accoda i lavori dietro un'API locale, e un timer di sistema esegue la verifica settimanale.

Contro un backup dedicato

Nel banco con inserzioni il profilo di backup risparmia quanto restic (dal 97 al 111 % del suo risparmio), con la copia singola entro un decimo di punto.

BACKUP E VERSIONI: SCENARI DA SCRIPT DETERMINISTICO, RIPRODUCIBILI DA TERZI; LUGLIO 2026

SCENARIO	TSR	7-ZIP AL MASSIMO
Database SQLite da 408 MB, due versioni dopo dodici modifiche	100,0 MB (0,90 GB di memoria)	109,8 MB (8,30 GB)
Log che cresce in coda	29,2 MB in 26,2 s	35,3 MB in 136,8 s
Immagine disco modificata montandola	247,1 MB	427,6 MB
Cartella da 389,5 MB più il suo backup completo	36,6 % (il secondo costa 2.190 byte)	74,8 %

06 Le prove, e il metodo che le governa

Un benchmark vale quanto le sue condizioni. Le regole qui sotto governano ogni cifra di questo documento, e valgono da mesi per ogni misura del progetto.

Quattro regole

- **Le dimensioni sono deterministiche.** La dimensione di un archivio dipende soltanto dall’input e dal profilo: una corsa basta, e ripetuta altrove dà gli stessi byte.
- **I tempi richiedono disciplina.** La deriva termica vale fino al 20 %: dove un tempo sostiene un’affermazione la misura è un confronto A/B interfogliato su tre giri, e conta il minimo; i tempi delle tabelle di quadro sono a corsa singola.
- **Chi costruisce il corpus decide il risultato.** Quattro criteri eseguiti da uno script: nessun file oltre il dieci per cento della massa; nessun materiale già usato nei corpora di misura o nell’addestramento del codec; fonti pubbliche con impronta verificata; nessuna selezione per tipo dentro una fonte. La lezione è stata pagata in casa: un corpus interno usato fino a inizio agosto 2026 mostrava ventidue punti di distacco, e sui corpora rifatti così il distacco vale circa tre. Corpus ritirato, misura rifatta, episodio scritto.
- **I rivali al loro meglio, e i verdetti negativi pubblicati.** Dieci strumenti ai livelli massimi documentati, compresi i compressori più forti sul solo rapporto (zpaq, bsc, PPMd); i piazzamenti si dichiarano contro tutti, e le idee respinte hanno una pagina nel dossier di benchmark.

Le quattro macchine

Apple M5 Pro, 15 core (macOS, riferimento dei tempi); NVIDIA GB10, venti core ARM (Linux arm64); AMD Ryzen 9 5900X, 12 core, in Windows 10 e in WSL2 Ubuntu 22.04 (x86-64).

I dieci strumenti

7-Zip -mx9 e la variante PPMd (o32 mem2g), zpaq -m5, xz -9e, zstd --ultra -22, brotli -q11, bzip2 -9, bsc -b100 -e2. Sugli scenari d’uso: zip -9, 7-Zip -mx9 e -mx5, zstd -19 --long=27, xz -9.

La frase che i numeri sostengono

Primo sui contenitori compressi; 7-Zip, xz, zstd, brotli e bzip2 battuti su testo, dati misti e documenti. Sul testo puro zpaq, bsc e PPMd stanno davanti: sta scritto qui perché è la prima cosa che un revisore misura.

I risultati sui corpora

CORPUS	TSR MASSIMO	MIGLIOR RIVALE	PIAZZAMENTO
documenti-ufficio (1.322 file, 199,5 MB)	47,888 %	zpaq 51,98 %	primo su dieci
documenti-pubblici (Govdocs1, 980 file)	48,598 %	zpaq 47,51 %	secondo
Silesia (211,9 MB)	20,124 %	zpaq 18,81 %	secondo
enwik8 (100 MB di testo)	21,975 %	zpaq 19,63 %	quarto, dietro zpaq, bsc e PPMd
png-grafica (11.356 PNG ottimizzati)	53,373 %	zpaq 52,84 %	terzo
sorgenti (13.663 file piccoli, su tar)	14,311 %	—	16,555 % sulla somma dei file
galleria JPEG (5 fotografie)	81,840 %	generici ≈ 100 %	−18,2 %; uno specialista ≈ −22 %

Profilo massimo, 8 settembre 2026, Apple M5 Pro; rivali ai livelli massimi sulla stessa macchina; denominatore il tar del corpus. 7-Zip, xz, zstd, brotli e bzip2 battuti su testo, dati misti e documenti.

Nove scenari d'uso simulati

Nove cartelle che somigliano a quelle di una persona, da 3 a 12 GB l'una, costruite da fonti pubbliche con licenza per file (44,9 GiB e 195.753 file in tutto) e misurate su quattro macchine. Qui il Mac a GPU spenta: TSR con la mediana di tre giri, i rivali a un giro al loro meglio. Rapporto in percentuale sui byte grezzi, più basso è meglio; in grassetto il migliore per riga.

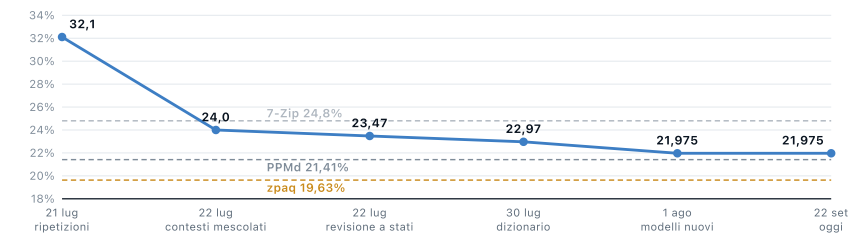
NOVE SCENARI D'USO: RAPPORTO IN PERCENTUALE SUI BYTE GREZZI, APPLE M5 PRO, 12-14 SETTEMBRE 2026

PERSONA	TSR MASSIMO	ZIP -9	7-ZIP -MX9	ZSTD -19 --LONG	XZ -9
Medico	34,71	67,32	46,62	45,47	57,10
Famiglia	83,76	98,02	97,41	97,57	97,63
Creativo	69,51	84,44	79,61	80,75	79,92
Studente	69,39	80,35	74,71	75,07	75,19
Legale	61,91	69,98	62,07	62,76	62,50
Videomaker	97,18	99,45	98,68	98,81	98,77
Ufficio	52,73	61,11	51,02	51,52	51,41
Tecnico IT	54,51	57,15	51,57	51,76	54,51
Sviluppatore	43,84	52,71	36,80	38,47	39,11

Primo dove lavorano i filtri (immagini mediche, JPEG del telefono, PNG, PDF) con margini da 5 a 13,7 punti sul migliore degli altri; dietro 7-Zip in modalità solida dove i file sono tanti e piccoli (sviluppatore, ufficio), perché comprime dentro i blocchi di un file e mai attraverso i file. Il profilo bilanciato produce sempre un archivio più piccolo di zip -9; il veloce finisce sempre prima di 7-Zip al livello 5, da 3 a 38 volte.

Da dove siamo partiti

enwik8, profilo massimo: come è sceso il rapporto (più basso è meglio)



Il codec è nato il 21 luglio 2026 e perdeva contro gzip. Dal 1° agosto il numero di enwik8 è fermo: il lavoro successivo è andato sui formati già compressi, sulla velocità e sull'affidabilità. Ogni passo ha un verbale con la misura, e i passi respinti stanno accanto a quelli accolti.

Affidabilità misurata

Suite di **632 casi** in 49 gruppi; verde sulle quattro piattaforme allo stesso commit, il 27 settembre 2026.

49 vettori di conformità letti da tre lettori indipendenti; 13.649 file di 1.540 tipi da quattro sistemi: estrazione identica, zero arresti, zero non determinismi.

2.752 spegnimenti a metà scrittura senza un archivio perso; 11,77 milioni di esecuzioni di fuzzing registrate, zero difetti.

Scala: un milione e cinque milioni di file dentro il budget di memoria (aprire cinque milioni di file costa 19 MB); un archivio da 111 GB e 370 mila file in una corsa sola, con la verifica profonda verde e mille file a caso identici.

07 Perimetro dichiarato

Questa pagina esiste perché chi elenca soltanto i propri vantaggi costringe il lettore a scoprire il resto da solo. Qui c'è il resto, con i numeri.

ASPETTO	STATO DICHIARATO ALLA DATA DI QUESTA EDIZIONE
Testo puro	Su enwik8 zpaq, bsc e PPMd stanno davanti (19,63, 20,80 e 21,41 % contro 21,975); su Silesia e Govdocs1 davanti resta il solo zpaq. TSR è primo dove lavorano i filtri, ed è lì che il prodotto lavora.
Media già compressi	MP3, AAC e fotografie non trattate: recupero dell'1–2 %. Film H.264: 2–3,7 %, misurato su file personali non ridistribuibili. Su una videoteca TSR vale per l'accesso e per l'integrità.
File tanti e piccoli	Blocchi dentro un file, mai attraverso i file: sugli alberi di sorgenti ed e-mail 7-Zip in modalità solida sta davanti di 2–7 punti. È il prezzo dell'accesso casuale.
Costo del rapporto massimo	Il profilo massimo impegna la macchina per decine di secondi ogni cento megabyte per core; la fluidità viene dal parallelismo, e l'utente sceglie il tempo.
Velocità sull'hardware di destinazione	Quattro macchine dichiarate. Sul Ryzen 9 5900X il profilo massimo fa 4,0 MiB/s aggregati: i 50 MB/s su CPU a 16–24 core restano fuori portata col codec attuale, e la velocità si dichiara per profilo.
Memoria	Senza un tetto esplicito il budget è una stima, che gli alberi pieni di foto e video possono superare; il tetto chiesto è un contratto rispettato (1,79 GB con un tetto di 2), pagando fino a tre volte il tempo. La riga di comando dice quale dei due è in uso.
Stime	Sono prudenti. I profili a codec fisso restano prudenti di circa tre volte sul tempo, perché l'efficienza del parallelismo è tarata sulla macchina di sviluppo.
JPEG e PNG	JPEG: 18,2 % contro il 22 % circa di uno specialista. PNG: soltanto i file a compressione classica; la grafica ottimizzata moderna viaggia com'è, e su 11.356 file di quel tipo TSR è terzo.
Deduplicazione	Una modifica sparsa in un file grande riscrive ogni blocco toccato: 13,8 MB cambiati a caso in 1,3 GB costano 101 MB (un backup dedicato paga lo stesso).
Prove di scala e di durata	Fatte fino a cinque milioni di file e a 111 GB in una corsa sola. Le ventiquattro ore di servizio continuo hanno il criterio di accettazione scritto e restano da eseguire.
Certificazioni e attestazioni	Nessuna certificazione; test di penetrazione e revisione crittografica indipendenti da commissionare. Gli artefatti che li abilitano (distinta dei componenti, evidenze, politica di sicurezza) sono pronti.
Firma del codice	Su macOS firma ad hoc; i certificati di Apple e di Windows sono attesi, e fino ad allora i sistemi avvertono al primo avvio.
Canale di rilascio	Versione 0.10.2, «beta dimostrativa»: otto pacchetti e due immagini provati, e a parte una beta pubblica; un pilota presso terzi resta da fare.

Che cosa resta fuori dalla promessa del formato

- Un archivio nuovo su un programma vecchio si rifiuta, dicendolo: la promessa è che il programma nuovo legga ogni archivio vecchio.
- Gli stessi byte per sempre a parità di ingresso fra versioni diverse del programma: un codec nuovo può vincere l'elezione e cambiare i byte scritti. Garantiti sono la lettura e il contenuto estratto identico; dove i byte devono coincidere (fra architetture, con la stessa versione) lo dice la misura.
- La cifratura lascia visibili la dimensione dell'archivio e il numero di estensioni, e il rollback dell'intero file resta possibile a chi possiede il supporto.
- I metadati arrivano dove il filesystem li accetta; l'estrazione riesce comunque.

08 Verificare da sé, e adottare

Tre verifiche alla portata di chiunque, senza fidarsi di questo documento; poi in che forma il programma arriva.

Tre verifiche

- 1. I corpora pubblici.** enwik8 (Large Text Compression Benchmark) e Silesia si scaricano; i rivali si eseguono ai livelli massimi standard (7z -mx9, xz -9e, zstd --ultra -22, brotli -q11, bzip2 -9, zpaq -m5). La dimensione dell'archivio TSR è deterministica: una corsa basta, e coincide al byte con quella dichiarata.
- 2. Il contratto dell'acceleratore grafico.** tsr gpu-check esegue la ricerca delle ripetizioni sulla GPU e sulla CPU della macchina e confronta i risultati posizione per posizione.
- 3. Il proprio dato.** Trenta giorni di prova con gli script di confronto inclusi: la versione per le aziende gira nella loro rete, senza comunicazioni verso l'esterno, e alla fine resta una tabella di numeri propri. Un archivio qualunque si verifica per intero con tsr t --profonda.

In che forma arriva

Un'applicazione grafica per macOS, Windows e Linux (trascinamento, coda di lavori, anteprima, estrazione selettiva, verifica e riparazione); uno strumento a riga di comando con uscita JSON e nove codici di uscita, gli stessi della libreria C libtsr con cui si integra in altri programmi; otto pacchetti (zip per macOS e Windows, tarball, .deb e .rpm per x86-64 e ARM64) e due immagini per contenitori, una delle quali da 2,92 MB senza un sistema operativo dentro. Ogni pacchetto porta la distinta dei componenti del binario (SBOM), la politica di sicurezza, la licenza e le attribuzioni; le dipendenze sono tutte permissive, zero copyleft.

Su un server

Servizio di streaming con segreto d'accesso obbligatorio fuori dall'interfaccia locale; demone con coda di lavori e una radice che recinta i percorsi; verifica settimanale con un timer di sistema; unità systemd irrigidite; un'immagine per contenitori con dentro il solo binario statico.

Documenti collegati

TSR-COM-01 · Dossier commerciale
 TSR-TEC-02 · Dossier tecnico, con accordo di riservatezza
 TSR-PUB-03 · Scheda pubblica
 TSR-LEG-04 · Quadro contrattuale
 TSR-BEN-05 · Dossier di benchmark
 TSR-2P-07 · In due pagine
 Ogni documento anche in inglese
 SECURITY.md · la politica di sicurezza, dentro ogni pacchetto

Contatti

Matteo Luigi Feroldi
 Via Carlo Cattaneo 18/1
 20055 Vimodrone (MI), Italia
 P. IVA 14667380969

matteo@feroldi.cloud
 +39 388 8630382

```
tsr c archivio.tsr cartella/ -t lento          # livello a tempo; -t 300 per un budget in secondi
tsr c archivio.tsr cartella/ --max --stima     # codec fisso, con le stime prima di partire
tsr t archivio.tsr --profonda                 # verifica, ricostruendo anche i file smontati
tsr serve archivio.tsr                        # HTTP con Range e WebDAV, sola interfaccia locale
tsr mount archivio.tsr /mnt/a                 # Linux: l'archivio come cartella
tsr scrub archivio.tsr --banda 20M --cpu 25    # la verifica piano, che riprende da dove era
```

Un punto percentuale di rapporto si guadagna una volta e rende su ogni copia del dato, per tutta la sua vita.

Glossario e riferimenti

I termini usati in questo documento, nel senso in cui TSR li usa, e le fonti pubbliche delle misure.

Rapporto	La dimensione dell'archivio in percentuale dell'originale; più basso è meglio. Un «punto» è un punto percentuale.	Guardia dell'incomprimibile	Il campione che riconosce i dati già compressi o casuali prima di eseguire i codec.
Blocco	L'unità di compressione, da 2 a 16 MiB secondo il profilo, sempre dentro un file; ogni blocco si comprime, si verifica e si legge da solo.	Determinismo	Lo stesso input e lo stesso profilo producono gli stessi byte, su macchine diverse.
Profilo	L'insieme delle scelte di una compressione: a tempo (cinque livelli o un budget in secondi) oppure a codec fisso (veloce, bilanciato, backup, massimo).	Impronta	L'hash BLAKE3 di un blocco, di un file o dell'indice: dice se il contenuto è quello di origine.
Calibrazione	La misura di velocità e rapporto di ogni codec sui file veri e sulla macchina in uso, prima di partire.	Record di riparazione	La parità Reed–Solomon in coda all'archivio, che ripara un danno entro la quota scelta.
Filtro	La riscrittura reversibile di un file già compresso, dimostrata su ogni file prima di scrivere.	Vettore di conformità	Un archivio congelato con la dichiarazione di ciò che deve venir fuori a leggerlo; quarantanove nel progetto.
Contesti mescolati	Context mixing: la famiglia di codec che predice ogni bit da più modelli e ne pesa le opinioni. La corsia forte di TSR.	Scrub	La verifica che va piano, si ferma e riprende da dove era.
Elezione	Sui budget ampi il codec prova più candidati sullo stesso blocco e tiene il più corto.	Contenitore DEFLATE	Un file che contiene flussi compressi col DEFLATE del 1993: ZIP, documenti Office, JAR, gzip, PNG, gli stream dei PDF.

Riferimenti

- **enwik8**: i primi cento milioni di byte della Wikipedia inglese, dal Large Text Compression Benchmark, dove sono pubblicati anche i risultati dei compressori più forti.
- **Silesia Corpus**: dodici file per 211,9 MB (testo, binari, database, immagini mediche), il corpus misto di riferimento del settore.
- **Govdocs1**: documenti reali raccolti da siti pubblici, resi disponibili da Digital Corpora; qui il thread 000 intero.
- **Rivali**: 7-Zip (anche nella variante PPMd), xz, zstd, brotli, bzip2, zpaq, bsc; sugli scenari d'uso anche zip. Tutti ai livelli massimi documentati, con i parametri scritti accanto.
- **Primitive crittografiche**: XChaCha20–Poly1305 per la cifratura autenticata, Argon2id per la derivazione della chiave, BLAKE3 per le impronte, Reed–Solomon su GF(2¹⁶) per la parità.
- **Date delle misure**: rapporti dell'8 settembre 2026; scenari d'uso del 12–14 settembre (la colonna del profilo veloce rimisurata il 22); prove di scala del 18–20; pacchetti e ricostruzione parallela del 22 settembre, suite del 23 settembre 2026.

Come leggere le percentuali

Ogni rapporto è la dimensione dell'archivio divisa per quella dell'originale: 47,888 % vuol dire che l'archivio occupa meno della metà. I rapporti si confrontano per differenza (i «punti»); un guadagno del 18,2 % sui JPEG è la riduzione della dimensione rispetto al file di partenza.

Dove stanno i dettagli

Il dossier di benchmark porta corpora, comandi e macchine di ogni cifra; il dossier tecnico, consegnato con accordo di riservatezza, il funzionamento interno del codec e del formato; la scheda pubblica il quadro in otto pagine.



Time Space Reducer — comprimere di più dove gli altri si fermano, in un tempo che si sceglie, con un archivio che è una funzione del dato.

TITOLARE

Matteo Luigi Feroldi
Via Carlo Cattaneo 18/1
20055 Vimodrone (MI), Italia
P. IVA 14667380969

CONTATTI

matteo@feroldi.cloud
+39 388 8630382

RIPRODUZIONE DELLE MISURE PUBBLICHE

enwik8 — Large Text Compression Benchmark
Silesia — Silesia Corpus
Rivali ai livelli massimi standard

TSR — Time Space Reducer e il formato d'archivio .tsr sono software proprietario. Copyright © 2026 Matteo Luigi Feroldi. Tutti i diritti riservati. I marchi citati appartengono ai rispettivi titolari e sono nominati unicamente per identificare i prodotti messi a confronto. Le misure riportate sono state eseguite sulle macchine e sui corpora dichiarati accanto a ciascuna cifra; risultati su hardware e dati diversi possono variare. Documento TSR-WHP-06, edizione del 27 settembre 2026, riferita alla versione 0.10.2 del programma e al formato 1.0 (contenitore v8).