On the Value of Multiple Read/Write Streams for Data Compression

Travis Gagie

University of Eastern Piedmont

CPM '09

Models of disks:

- external-memory I/O
  [Aggarwal & Vitter, 1988]
- cache-oblivious
  [Frigo *et al.*, 1999]
- read/write streams
  [Grohe, Koch & Schweikardt, 2005]

Read/write streams:

- sequential access
- sublinear memory
  (e.g., polylogarithmic)
- multiple passes
  (e.g., polylogarithmic)
- ability to change the data
- possibility of multiple streams

	1 stream	$\mathcal{O}(1)$ streams
e.g., sorting (see [Sch07])	NO	YES
universal compression	YES	YES
grammar-based compression	NO	?
entropy-only compression	NO	YES

Theorem (Gupta, Grossi & Vitter, 2008)

It takes $\mathcal{O}(|s|)$ *time in the RAM model to store a string* s *in* $|s|H_k(s) + \mathcal{O}(\log |s|)$ *bits.*

Corollary

We can achieve universal compression in the standard streaming model.

Proof.

1. process s in blocks of $\log^c |s|$ characters
2. apply GGV's algorithm to each block in turn
3. redundancy is $\mathcal{O}\left(\frac{n \log \log |s|}{\log^c |s|}\right)$

$\square$

Introduction

Universal compression

Grammar-based compression

Entropy-only compression

Theorem (Rytter, 2003; Charikar *et al.*, 2005)

In the RAM model, we can approximate the smallest-grammar problem with $|\mathrm{APPROX}| \leq |\mathrm{OPT}|^2$.

Lemma

We can compute any substring of s from

- *the length of the substring,*
- *the machine's configurations when it reaches and leaves the part of the stream that initially holds that substring,*
- *all the output it produces while over that part.*

Theorem

With 1 stream, we cannot approximate the smallest-grammar problem with $|\mathsf{APPROX}| \leq |\mathsf{OPT}|^{\mathcal{O}(1)}$.

Proof.

1. m memory and p passes, $mp = \log^{\mathcal{O}(1)} |s|$
2. $s = t^r$, $|t| = (mp)^{1+\epsilon}$
3. t is a randomly chosen string (incompressible w.h.p.)
4. $|\mathrm{OPT}| = \mathcal{O}\left(|t| + \log r\right) \subset \log^{\mathcal{O}(1)} |s|$
5. by lemma, output for any copy of t is $\Omega(|t|)$ bits
6. $|\mathrm{APPROX}| = \Omega(|s|)$

□

Theorem

With 1 stream, we cannot achieve entropy-only compression, i.e., a bound of the form $\lambda|s|H_k^*(s) + g_k$.

Proof.

1. m memory and p passes, $mp = \log^{\mathcal{O}(1)} |s|$
2. $s = t^r$, $|t| = 2^k = (mp)^{1+\epsilon}$
3. t is a randomly chosen kth-order De Bruijn cycle (compressible by at most a constant factor w.h.p.)
4. $|s| H_k^*(s) = \mathcal{O}\left(|t| \log r\right) \subset \log^{\mathcal{O}(1)} |s|$
5. by lemma, output for any copy of t is $\Omega(|t|)$ bits
6. output is $\Omega(|s|)$ bits

$\square$

Theorem (Manzini, 2001)

Using the Burrows-Wheeler Transform, move-to-front coding and arithmetic coding, we can achieve entropy-only compression.

Lemma (Ruhl, 2003 + Hardy, c. 1967 + tweak)

We can compute and invert the Burrows-Wheeler Transform using $\mathcal{O}(\log |s|)$ *bits of memory and* $\mathcal{O}\left(\log^2 |s|\right)$ *passes over 2 streams.*

Theorem

With 2 streams, we can achieve entropy-only compression.

	1 stream	$\mathcal{O}(1)$ streams
e.g., sorting (see [Sch07])	NO	YES
universal compression	YES	YES
grammar-based compression	NO	?
entropy-only compression	NO	YES