draft-ietf-bmwg-mlrsearch-03

IETF-116 Yokohama, BMWG Meeting

Authors: Maciek Konstantynowicz, Vratko Polák

Work Status

- draft-ietf-bmwg-mlrsearch-03 posted on 9th of November 2022 (post IETF-115 deadline)
  - Aligned terminology with RFCs: 2242, 2285, 2544.
- Added functionality to handle noisy and inconsistent measurements even better
  - This required a change in logic
    - A search "phase" has been split into search "target" and load "selector".
    - Load "classification" added, as it is no longer trivial.
    - More details on the new logic are in this presentation.
- Mlrsearch draft
  - Update draft to incorporate new load classification and target attributes.
  - Terminology to be finalized.
- BMWG review
  - Scrutiny and validation of listed problems.
  - Review of MLRsearch draft once updated with new logic (per this presentation).

Problems

- Long Search Duration
  - Vanilla bisection is too slow
- DUT within SUT
  - DUT – software program processing packets, device of interest
  - SUT – server hardware and operating system, server resources shared
  - DUT is effectively ”nested” within SUT
  - SUT performance is a spectrum, the higher end is “noiseless” and more stable
- Repeatability and Comparability
  - Need for search procedure that reports more stable results across iterations and environments
- Throughput with Non-Zero Loss
  - Searching for multiple loss ratio goals helps to describe SUT performance better
  - Non-Zero loss performance is more stable
- Inconsistent Trial Results
  - Is a zero-loss trial a throughput if there was non-zero-loss trial with smaller load?

Multiple Loss Ratio search: Motivation and Objectives

- Benchmarking methodology extending network throughput search specified in RFC 2544
- Focus on software networking applications running on shared COTS compute servers
- Address problems observed while testing software networking data planes
  - Long search duration
  - DUT within SUT
  - Repeatability and comparability
  - Throughput with non-zero Loss
  - Inconsistent trial results
- Main objectives:
  - Minimize total search duration
  - Search for multiple loss ratios
  - Improve results repeatability and comparability

High level description

- After pre-initialization (using FRMOL), the search enters its main loop.
- In a main loop iteration, the algorithm decides which offered load and trial duration to use, then performs one trial measurement.
  - The algorithm tracks multiple **target**s at once.
  - There is a load **selector** focusing on each target.
  - (More details on targets and selectors are below and in the following slides.)
  - Each load selector may propose one **candidate** load for this iteration.
  - The smallest candidate load is the one used for the next measurement.
  - If there are no candidates, the search is done.
  - If multiple selectors propose the same load with different durations, the longest duration is used.
- Load selector decisions are affected by trial results only via a **load classification**.
  - An offered load is classified as either an **upper bound**, **lower bound**, or **unknown**.
    - With respect to specific target. Other targets may classify the same load differently.
    - Results of all trials at that offered load contribute to the load classification.
- For each *final* target, the **conditional throughput** is the average forwarding rate (across *good longer* trials) measured at the highest (offered load classified as) lower bound.

Definition of Target

- A target is a tuple of scalar attributes.
  - 5-tuple for load selector purposes.
  - Effectively a 4-tuple for load classification purposes.
  - The 4-tuple can also classify a single trial result as good/bad and shorter/longer.
- Attributes:
  - **Loss ratio**:
    - Trial loss ratio is the count of frames lost divided by the count of frames transmitted (towards SUT).
    - If a trial loss ratio is larger than the target loss ratio, the trial result is considered **bad**. Otherwise, it is considered **good**.
  - **Main trial duration**:
    - Load selector will propose this duration for its candidate load.
    - Trials with duration smaller than this are considered **shorter**, otherwise they are considered **longer**.
    - User chooses this value based on DUT buffers, and on what frequency of "noise spikes" can be tolerated.
  - **Min duration sum**:
    - Load classification expects at least this amount of time to be spent measuring this load in total.
    - A load can be classified when smaller than this sum was spent, if the remaining seconds cannot change the classification even in best/worst case.
  - **Exceed ratio**:
    - What portion of the duration sum can consist of bad trial seconds while still being classified as lower bound.
    - Zero for RFC 2544 compatibility, non-zero to get results equivalent to Binary Search with Loss Verification.
  - **Width**:
    - Target is **achieved** when the highest lower bound is no more than this far from the lowest upper bound.
    - No effect on load classification.

Relations between targets and selectors

- Each target may (or may not) have an associated **preceding** target.
  - Targets form chains (so you can talk about **initial** targets, middle targets and **final** targets).
  - Final target in a chain is directly derived from user inputs.
  - Other targets are derived indirectly.
  - Preceding target has wider width and smaller duration.
  - By achieving the preceding targets, the final targets are likely to be achieved faster.
- Load selector uses load classifications only with respect to its **current** target and (if available) its preceding target.
- Each selector is trying to achieve its current target.
  - Pre-initialization makes sure selectors for initial targets can make progress.
  - Non-initial targets are in a **waiting** state, until their preceding target is achieved.
  - Selector chooses a candidate load, unless it is waiting or **done**.

Load Classification Examples: Legend

- The upcoming table is for a single load, single target, different sets of trial results.
- The single target:
  - loss ratio = 0.0
  - main trial duration = 10s
  - duration sum = 100s
  - exceed ratio = 0.3 (30% of total seconds may come from bad trials)
- Classification relies on the following quantities, each is a sum of durations of trials:
  - GS (**Good** *Shorter*): each trial duration is *shorter* than 10s, each trial result is **zero loss**
  - BS (**Bad** *Shorter*): each trial duration is *shorter* than 10s, each trial result is **non-zero loss**
  - GL (**Good** *Longer*): each trial duration is *at least* 10s, each trial result is **zero loss**
  - BL (**Bad** *Longer*): each trial duration is *at least* 10s, each trial result is **non-zero loss**
- Each row in the table is one example, for the load of the last bisection
  - Lower/upper bound means the bisection is done.
    Unknown means at least one more 10s trial is needed.
  - Some rows can be seen as "continuations" of other rows (both rows can happen in a single search), but in general each row comes from a different search.
  - Some of GS and BS may have been proposed by an unrelated target (still the same load), with main trial duration 1s. (This example target does not imply multiple 1s trials.)

Load Classification Examples: Table

Row	GS[s]	BS[s]	GL[s]	BL[s]	Classification	Explanation
1	0	0	70=7*10	0	Lower bound	Even if all remaining 30s are bad, 70:30 is matching the exceed ratio.
2	0	0	60=6*10	0	Unknown	In the worst case the missing 40s can all be bad.
3	0	0	0	30=3*10	Unknown	The remaining 70s may be good.
4	0	0	0	40=4*10	Upper bound	Not enough remaining seconds to match the exceed ratio
5	0	1=1*1	0	30=3*10	Upper bound	The 1s counts as “bad” even if short (you cannot un-loose packets), so there is only 69s remaining, not enough to match the exceed ratio.
6	10=10*1	0	60=6*10	0	Unknown	If some of short trials were measured for longer, they might become bad, so the short trials do not count and there is enough seconds remaining for bad long trials.
7	7=7*1	3=3*1	0	30=3*10	Unknown	GS and BS "cancel each other", not counting them to the 100s limit.
8	7=7*1	4=4*1	0	30=3*10	Upper bound	1s of BS remains after cancelation, counting that does not leave enough seconds remaining to match exceed ratio.

No shortcut towards Lower bound, you need enough GL in any case.
BS can be a shortcut towards Upper bound, if there is not enough GS to cancel it.

THANK YOU !

draft-ietf-bmwg-mlrsearch-03

IETF-116 Yokohama, BMWG Meeting

Authors: Maciek Konstantynowicz, Vratko Polák

[backup slide] Load Selector Example

- Legend:
  - CL: Tightest lower bound for the current target.
    - Arbitrary integer units, you can imagine Mpps.
  - CU: Tightest upper bound for the current target.
  - PL: Tightest lower bound for the preceding target.
  - PU: Tightest upper bound for the preceding target.
  - Candidate: The selected load, it gets measured if it is lowest among selectors.
    - The newly measured value is in bold.
  - Action: Short description of how the candidate load is chosen.
  - Current target width is 1, preceding target width is 2.

CL	CU	PL	PU	Action	Candidate	Explanation
8	-	10	13	Wait	-	Preceding target,2 ,is not achieved yet.
8	-	10	**12**	Halve	11	This way only the remaining bound has to be refined.
11	-	11	12	Refine	12	Does the old PU become the new CU?
12	-	12	13	Extend up	14	No, jump up with increased width.
12	**14**	12	13	Bisect	13	We have CL and CU but they are too far apart.
13	14	13	14	Done	-	The current target has been achieved.

[backup slide] Load classification: Formulas

- 1. Compute the four duration sums (GS, BS, GL, BL) from trial results so far
- 2. Compute EBS (Excess Bad Short duration sum): EBS = BS – GS*ER/(1-ER)
  - ER is the Exceed Ratio target attribute
  - E.g.: EBS = BS – GS*3/7 when ER = 0.3
- 3. Compute EBL (Effective Bad Long duration sum): EBL = BL + max(0, EBS)
- 4. Compute ETL (Effective Total Long duration sum): ETL = max(MDS, GL + EBL)
  - MDS is the Min Duration Sum target attribute, e.g. 100s in the example table
- 5. Compute OXR (Optimistic Exceed Ratio): OXR = EBL / ETL
- 6. Compute PXR (Pessimistic Exceed Ratio): PXR = (ETL - GL) / ETL
- 7. If both ratios are:
  - Above the exceed ratio, the load is classified as Upper bound.
  - Below (or equal to) the exceed ratio, the load is classified as Lower bound.
  - Else the load is classified as Unknown.