draft-ietf-bmwg-mlrsearch-00

IETF-110 Online BMWG Meeting

Authors: Vratko Polák, Maciek Konstantynowicz

Presented by: Vratko Polák

Draft Status

- No official changes since draft-vpolak-mkonstan-bmwg-mlrsearch-03 (expired September 7, 2020).
- Draft adopted by BMWG => draft-ietf-bmwg-mlrsearch-00.
- Several changes prepared (presented here), but new draft version is not ready yet.
- More reviews and comments are welcome.

- MLRsearch continues to be used in LFN FD.io CSIT open-source benchmarking projects to execute 1000s of benchmarking test runs as part of FD.io CSIT CI/CD pipeline.
  - NDRPDR Trending Graphs (executed weekly): https://docs.fd.io/csit/master/trending/index.html
  - FD.io release benchmarks:
    - VPP: https://docs.fd.io/csit/rls2009/report/vpp_performance_tests/packet_throughput_graphs/index.html
    - DPDK Apps: https://docs.fd.io/csit/rls2009/report/dpdk_performance_tests/packet_throughput_graphs/index.html

Recap: IETF-109 slide

MLRsearch Sample Implementation

- A working implementation of MLRsearch is in Linux Foundation FD.io CSIT project.
  - Used for continuous measurements of NDR and PDR rates of:
    - FD.io VPP
    - DPDK L3fwd
    - DPDK Testpmd
  - Sample throughput results:
    - https://docs.fd.io/csit/rls2101/report/vpp_performance_tests/packet_throughput_graphs/index.html
  - General project info:
    - https://wiki.fd.io/view/CSIT
    - https://git.fd.io/csit/
- MLRsearch Python package (older version) published on PyPI:
  - https://pypi.org/project/MLRsearch/

Recap: IETF-105 slide

Overview: Multiple Loss Ratio search (MLRsearch)

- MLRsearch discovers multiple packet throughput rates in a single search
  - With each rate associated with a distinct Packet Loss Ratio (PLR) criteria
- Provides much shorter execution times for cases when multiple rates need to be found:
  - For example in NFV benchmarking to discover both NDR and PDR throughput
    - NDR: Non-Drop Rate with PLR=0, zero packet loss
    - PDR: Partial-Drop Rate with PLR>0, non-zero packet loss
  - Instead of running separate binary searches for NDR and PDR.

Recap: IETF-105 slide

Overview: Multiple Loss Ratio search (MLRsearch)

- MLRsearch execution time gets reduced even further
  - By using shorter trial durations in the intermediate steps
  - With only the final measurements conducted at the specified final trial duration.
- MLRsearch is a packet throughput search algorithm suitable for deterministic systems
  - As opposed to probabilistic systems

MLRsearch is compatible with RFC2544.

Recap: IETF-107 slide

Example MLRsearch Run (Section 5.2.)

- Table on the right shows data from a real test run in CSIT, using the default input values as described in the draft.
- The first column is the MLRsearch phase.
- The second is the trial measurement performed
  - Aggregate bidirectional offered load in mega (10^6) packets per second, and trial duration in seconds.
- Each of last four columns show one bound as updated after the measurement
  - Duration truncated to save space.
- Loss ratio is not shown, but invalid bounds are marked with a plus sign.
- Black bold font signifies changed values.
- Blue bold font signifies results of the search.

Phase	Trial	NDR lower	NDR upper	PDR lower	PDR upper
init.	37.50 **1.00**	N/A	**37.50** 1.	N/A	**37.50** 1.
init.	10.55 1.00	**+10.55** 1.	37.50 1.	**+10.55** 1.	37.50 1.
init.	9.437 1.00	**+9.437** 1.	**10.55** 1.	**+9.437** 1.	**10.55** 1.
int 1	6.053 1.00	**6.053** 1.	**9.437** 1.	**6.053** 1.	**9.437** 1.
int 1	7.558 1.00	**7.558** 1.	9.437 1.	**7.558** 1.	9.437 1.
int 1	8.446 1.00	**8.446** 1.	9.437 1.	**8.446** 1.	9.437 1.
int 1	8.928 1.00	**8.928** 1.	9.437 1.	**8.928** 1.	9.437 1.
int 1	9.179 1.00	8.928 1.	**9.179** 1.	**9.179** 1.	9.437 1.
int 1	9.052 1.00	**9.052** 1.	9.179 1.	9.179 1.	9.437 1.
int 1	9.307 1.00	9.052 1.	9.179 1.	9.179 1.	**9.307** 1.
int 2	9.115 **5.48**	**9.115** 5.	9.179 1.	9.179 1.	9.307 1.
int 2	9.243 5.48	9.115 5.	9.179 1.	**9.243** 5.	9.307 1.
int 2	9.179 5.48	9.115 5.	9.179 5.	9.243 5.	9.307 1.
int 2	9.307 5.48	9.115 5.	9.179 5.	9.243 5.	**+9.307** 5.
int 2	9.687 5.48	9.115 5.	9.179 5.	**9.307** 5.	**9.687** 5.
int 2	9.495 5.48	9.115 5.	9.179 5.	9.307 5.	**9.495** 5.
int 2	9.401 5.48	9.115 5.	9.179 5.	9.307 5.	**9.401** 5.
final	9.147 **30.0**	9.115 5.	**9.147** 30	9.307 5.	9.401 5.
final	9.354 30.0	9.115 5.	9.147 30	9.307 5.	**9.354** 30
final	9.115 30.0	**+9.115** 30	9.147 30	9.307 5.	9.354 30
final	8.935 30.0	**8.935** 30	**9.115** 30	9.307 5.	9.354 30
final	9.025 30.0	**9.025** 30	9.115 30	9.307 5.	9.354 30
final	9.070 30.0	**9.070** 30	9.115 30	9.307 5.	9.354 30
final	9.307 **30.0**	**9.070** 30	**9.115** 30	**9.307** 30	**9.354** 30

Example MLRsearch old logic

- Table on the right shows fake data.
- The first column is the MLRsearch phase.
- The second is the trial measurement performed
  - Aggregate bidirectional offered load in mega (10^6) packets per second, and trial duration in seconds.
- Each of last four columns show one bound as updated after the measurement
  - Duration truncated to save space.
- Loss ratio is not shown, but invalid bounds are marked with a plus sign.
- **Bold** font signifies changed values.
- Blue font signifies results of the search.
- Red font highlights the inefficient decision.

Phase	Trial	NDR lower	NDR upper	PDR lower	PDR upper
init.	20.00 **1.00**	N/A	**20.00** 1.	N/A	**20.00** 1.
init.	16.00 1.00	**+16.00** 1.	20.00 1.	**+16.00** 1.	20.00 1.
init.	15.00 1.00	**+15.00** 1.	**16.00** 1.	**15.00** 1.	**16.00** 1.
int 1	13.00 1.00	**+13.00** 1.	**15.00** 1.	15.00 1.	16.00 1.
int 1	9.00 1.00	**9.00** 1.	**13.00** 1.	15.00 1.	16.00 1.
int 2	11.00 **5.48**	9.00 **1.**	**11.00 5.**	15.00 1.	16.00 1.
int 2	9.00 5.48	9.00 **5.**	11.00 5.	15.00 1.	16.00 1.
int 2	15.00 5.48	10.00 5.	11.00 5.	15.00 **5.**	16.00 1.
int 2	16.00 5.48	10.00 5.	11.00 5.	15.00 5.	16.00 **5.**
final	10.00 **30.0**	10.00 **30**	11.00 5.	15.00 5.	16.00 5.
final	11.00 30.0	10.00 30	11.00 **30**	15.00 5.	16.00 5.
final	15.00 30.0	10.00 30	11.00 30	+15.00 **30**	16.00 5.
final	13.00 30.0	10.00 30	11.00 30	**+13.00** 30	**15.00 30**
final	9.00 30.0	**+9.00** 30	**10.00** 30	**+9.00** 30	**13.00** 30
final	7.00 30.0	**7.00** 30	**9.00** 30	**7.00** 30	**9.00** 30
final	8.00 30.0	7.00 30	**8.00** 30	**8.00** 30	9.00 30

MLRsearch logic improvements

- Support configurable number of target loss ratios (not just NDR and PDR).
- Do not track just current bounds, track all measurement results.
- Maintain a “database” of results for each duration (phase).
    - Sort results in increasing intended load.
    - Calculate “effective loss ratio” to never decrease.
    - Database can be queried for tightest bounds and second tightest bounds.
- If the current duration database misses a convenient result, query the previous duration database.
- The code with improvements implemented: https://gerrit.fd.io/r/c/csit/+/30954

Example MLRsearch new logic

- Table on the right shows fake data.
- The first column is the MLRsearch phase.
- The second is the trial measurement performed
  - Aggregate bidirectional offered load in mega (10^6) packets per second, and trial duration in seconds.
- Each of last four columns show one bound as found after the measurement
  - Duration truncated to save space.
  - Only tightest bounds are shown.
  - Previous duration used if actual is missing.
- Loss ratio is not shown.
- **Bold** font signifies changed values.
- Blue font signifies results of the search.
- Green font highlights the efficient decisions.

Phase	Trial	NDR lower	NDR upper	PDR lower	PDR upper
init.	20.00 **1.00**	N/A	**20.00** 1.	N/A	**20.00** 1.
init.	16.00 1.00	N/A	**16.00** 1.	N/A	**16.00** 1.
init.	15.00 1.00	N/A	**15.00** 1.	**15.00** 1.	16.00 1.
int 1	13.00 1.00	N/A	**13.00** 1.	15.00 1.	16.00 1.
int 1	9.00 1.00	**9.00** 1.	13.00 1.	15.00 1.	16.00 1.
int 1	11.00 1.00	9.00 1.	**11.00** 1.	15.00 1.	16.00 1.
int 1	10.00 1.00	**10.00** 1.	11.00 1.	15.00 1.	16.00 1.
int 2	10.00 **5.48**	10.00 **5.**	11.00 1.	15.00 1.	16.00 1.
int 2	11.00 5.48	10.00 5.	11.00 **5.**	15.00 1.	16.00 1.
int 2	15.00 5.48	10.00 5.	11.00 5.	15.00 **5.**	16.00 1.
int 2	16.00 5.48	10.00 5.	11.00 5.	15.00 5.	16.00 **5.**
final	10.00 **30.0**	10.00 **30**	11.00 5.	15.00 5.	16.00 5.
final	11.00 30.0	10.00 30	11.00 **30**	15.00 5.	16.00 5.
final	15.00 30.0	10.00 30	11.00 30	**11.00 30**	**15.00 30**
final	13.00 30.0	10.00 30	11.00 30	11.00 30	**13.00** 30
final	12.00 30.0	10.00 30	11.00 30	11.00 30	**12.00** 30

MLRsearch future improvements

- Expansion coefficient for external search can be made configurable (CSIT uses 4 instead of 2).
- Even with new logic, it may be a good idea to use larger interval width goal for earlier phases.
  - This will make the logic more complicated, so we need data to prove speed improvement is worth it.
- Use uneven splits to avoid spending time on unneeded precision.
  - Example, if the current width is 3 times the goal:
  - Even splits result in 2 measurements, final width is three quarters of the goal.
  - 1:2 split needs 1.66 measurements on average, final width is equal to the goal.
  - The implementation has to be careful with respect to rounding errors.

THANK YOU !

draft-ietf-bmwg-mlrsearch-00

IETF-110 Online BMWG Meeting

Authors: Vratko Polák, Maciek Konstantynowicz

Presented by: Vratko Polák