

a4/conc_ll/run_on_queue.sh

```

1 #!/bin/bash
2
3 ## Job Name
4 #PBS -N run_conc_ll
5
6 ## Output and error of PBS (not the runs)
7 #PBS -o run_conc_ll.pbs_out
8 #PBS -e run_conc_ll.pbs_err
9
10 ## Sandman, serial queue, 64 threads available
11 #PBS -q serial
12 #PBS -l nodes=sandman:ppn=64
13
14 ## Maximum walltime (adjust if necessary)
15 #PBS -l walltime=01:00:00
16
17 ## Go to the directory where qsub was executed
18 # CHANGE THIS TO YOUR ACTUAL DIRECTORY
19 cd $HOME/a2/conc_ll
20
21 # --- Define Core Parameters ---
22 IMPLEMENTATIONS="serial cgl fgl opt lazy nb"
23 NTHREADS="1 2 4 8 16 32 64 128"
24 LIST_SIZES="1024 8192"
25
26 # Workloads: (Contains, Add, Remove)
27 # Format: "C_A_R"
28 WORKLOADS="100_0_0 80_10_10 20_40_40 0_50_50"
29
30 # Directory for results
31 OUTDIR="results_conc_ll"
32 mkdir -p "$OUTDIR"
33
34 # --- Helper Function to generate MT_CONF for thread pinning ---
35 # Generates a comma-separated list of logical core IDs.
36 # Assumes sandman has 64 logical cores (0-63).
37 # For N > 64, it cycles through the 64 available logical cores (oversubscription).
38 get_mt_conf() {
39     local N=$1
40     local CONFIG=""
41     local MAX_LOGICAL_CORES=64
42
43     for i in $(seq 0 $((N - 1))); do
44         # Core ID cycles through 0, 1, ..., 63, 0, 1, ...
45         local CORE_ID=$((i % MAX_LOGICAL_CORES))
46
47         CONFIG="${CONFIG}${CORE_ID}"
48         if [ $i -lt $((N - 1)) ]; then
49             CONFIG="${CONFIG},"
50         fi
51     done

```

```

52     echo "$CONFIG"
53 }
54
55 # --- Main Execution Loop ---
56 for IMPL in $IMPLEMENTATIONS; do
57     EXECUTABLE="./$x.$IMPL"
58
59     for S in $LIST_SIZES; do
60
61         for T in $NTHREADS; do
62
63             # For the serial implementation, only run T=1 (to establish baseline)
64             if [ "$IMPL" == "serial" ] && [ $T -gt 1 ]; then
65                 continue
66             fi
67
68             # --- MT_CONF Setting for Thread Pinning (pthreads) ---
69             if [ $T -gt 1 ]; then
70                 MT_CONF=$(get_mt_conf $T)
71                 export MT_CONF
72             else
73                 # Unset MT_CONF for single-threaded execution (T=1)
74                 unset MT_CONF
75             fi
76
77             echo "Running $IMPL: ListSize=$S, Nthreads=$T, MT_CONF=$MT_CONF"
78
79             for W in $WORKLOADS; do
80                 # Split the workload string (e.g., 100_0_0) into C, A, R variables
81                 IFS='_' read -r C A R <<< "$W"
82
83                 # Input arguments for the executable: <list_size> <contains_pct> <add_pct>
<remove_pct>
84                 ARGS="$S $C $A $R"
85
86                 # Output files for this run
87                 OUT="${OUTDIR}/conc_ll_${IMPL}_${S}${S}_T${T}_W${W}.out"
88                 ERR="${OUTDIR}/conc_ll_${IMPL}_${S}${S}_T${T}_W${W}.err"
89
90                 # Run the program:
91                 # - stdout → OUT
92                 # - stderr → ERR
93                 $EXECUTABLE $ARGS >"$OUT" 2>"$ERR"
94             done
95         done
96     done
97 done
98
99 echo "Execution finished. Results are in the $OUTDIR directory."
100

```