a3/run_on_queue.sh

```
#!/bin/bash

#PBS -N run_kmeans_locks
#PBS -o run_kmeans_locks.out
#PBS -e run_kmeans_locks.err
#PBS -l nodes=1:ppn=64
#PBS -l walltime=01:00:00

## How to submit (runs all locks × all thread configs on sandman):
##   qsub -q serial -l nodes=sandman:ppn=64 run_on_queue.sh
##
## Defaults (can be overridden via -v):
##   SIZE=32
##   COORDS=16
##   CLUSTERS=32
##   LOOPS=10

set -euo pipefail

# Work in the directory where qsub was executed (your a3 folder)
cd "${PBS_O_WORKDIR:-.}" || exit 1

# Fixed configuration required by the exercise (override with env if needed)
SIZE="${SIZE:-32}"
COORDS="${COORDS:-16}"
CLUSTERS="${CLUSTERS:-32}"
LOOPS="${LOOPS:-10}"

# Thread configurations to test
THREADS_LIST=(1 2 4 8 16 32 64)

# Lock variants (names as they appear in the binary targets)
LOCKS=(
  "nosync_lock"
  "pthread_mutex_lock"
  "pthread_spin_lock"
  "tas_lock"
  "ttas_lock"
  "array_lock"
  "clh_lock"
)

run_one() {
  local lock_name="$1"
  local threads="$2"
  local bin=""

  if [[ "${lock_name}" == "critical" ]]; then
    # OpenMP critical version
    bin="kmeans_omp_critical"
  else
```

```
    # Lock-based versions (built from omp_lock_kmeans.c + one lock object)
    bin="kmeans_omp_${lock_name}"
  fi

  if [[ ! -x "./${bin}" ]]; then
    echo "[WARN] Skipping lock='${lock_name}', threads=${threads}: binary '${bin}' not found"
    return
  fi

  # OpenMP settings
  export OMP_NUM_THREADS="${threads}"

  # Always use thread binding (affinity) as required
  local affinity=""
  for ((i=0; i<threads; i++)); do
    affinity+="${i} "
  done
  affinity="${affinity%% }"
  export GOMP_CPU_AFFINITY="${affinity}"

  # Result directory:
  #   benchmarks/<lock_name>/S32_N16_C32_L10_T8/
  local result_dir="benchmarks/${lock_name}/S${SIZE}_N${COORDS}_C${CLUSTERS}_L${LOOPS}_T${threads}"
  mkdir -p "${result_dir}"

  {
    echo "[run_on_queue] BIN=${bin}"
    echo "[run_on_queue] LOCK=${lock_name}"
    echo "[run_on_queue] OMP_NUM_THREADS=${OMP_NUM_THREADS}"
    echo "[run_on_queue] GOMP_CPU_AFFINITY=${GOMP_CPU_AFFINITY}"
    echo "[run_on_queue] Params: -s ${SIZE} -n ${COORDS} -c ${CLUSTERS} -l ${LOOPS}"
    echo "[run_on_queue] Result dir: ${result_dir}"
  } > "${result_dir}/meta.txt"

  echo "[INFO] Running lock='${lock_name}', threads=${threads}, bin='${bin}'"
  ./"${bin}" -s "${SIZE}" -n "${COORDS}" -c "${CLUSTERS}" -l "${LOOPS}" \
    | tee "${result_dir}/output.txt"
}

# 1) Run all lock implementations (omp_lock_kmeans.c + locks/)
for lock in "${LOCKS[@]}"; do
  for t in "${THREADS_LIST[@]}"; do
    run_one "${lock}" "${t}"
  done
done

# 2) Run the critical version (omp_critical_kmeans.c → kmeans_omp_critical)
for t in "${THREADS_LIST[@]}"; do
  run_one "critical" "${t}"
done


```