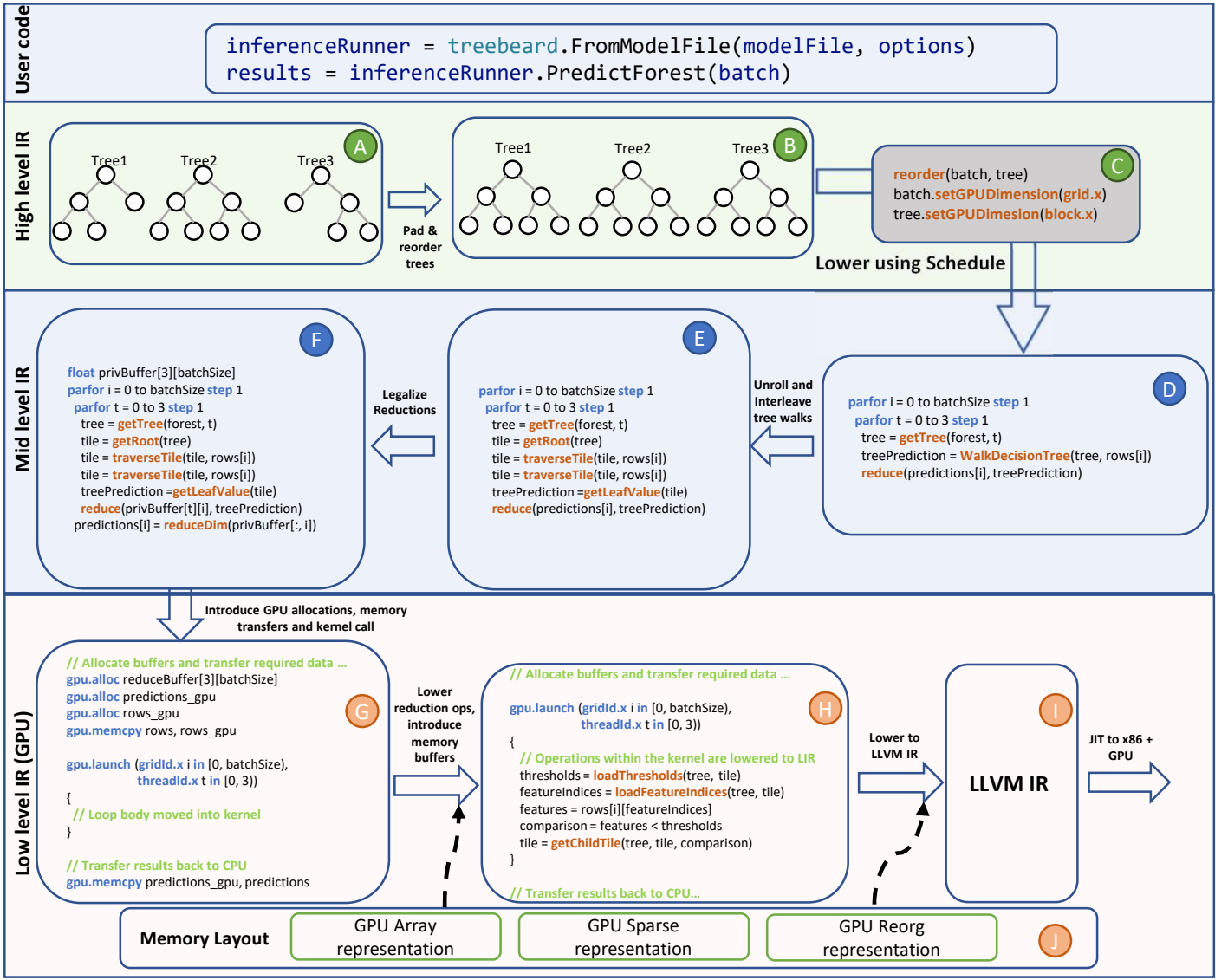
User code
inferenceRunner = treebeard.FromModelFile(modelFile, options)
results = inferenceRunner.PredictForest(batch)
High level IR
A
Tree1
Tree2
Tree3
Pad & reorder trees
B
Tree1
Tree2
Tree3
C
reorder(batch, tree)
batch.setGPUDimension(grid.x)
tree.setGPUDimesion(block.x)
Lower using Schedule
Mid level IR
D
parfor i = 0 to batchSize step 1
parfor t = 0 to 3 step 1
tree = getTree(forest, t)
treePrediction = WalkDecisionTree(tree, rows[i])
reduce(predictions[i], treePrediction)
Unroll and Interleave tree walks
E
parfor i = 0 to batchSize step 1
parfor t = 0 to 3 step 1
tree = getTree(forest, t)
tile = getRoot(tree)
tile = traverseTile(tile, rows[i])
tile = traverseTile(tile, rows[i])
treePrediction =getLeafValue(tile)
reduce(predictions[i], treePrediction)
Legalize Reductions
F
float privBuffer[3][batchSize]
parfor i = 0 to batchSize step 1
parfor t = 0 to 3 step 1
tree = getTree(forest, t)
tile = getRoot(tree)
tile = traverseTile(tile, rows[i])
tile = traverseTile(tile, rows[i])
treePrediction =getLeafValue(tile)
reduce(privBuffer[t][i], treePrediction)
predictions[i] = reduceDim(privBuffer[:, i])
Introduce GPU allocations, memory transfers and kernel call
Low level IR (GPU)
G
// Allocate buffers and transfer required data ...
gpu.alloc reduceBuffer[3][batchSize]
gpu.alloc predictions_gpu
gpu.alloc rows_gpu
gpu.memcpy rows, rows_gpu
gpu.launch (gridId.x i in [0, batchSize),
threadId.x t in [0, 3))
{
// Loop body moved into kernel
}
// Transfer results back to CPU
gpu.memcpy predictions_gpu, predictions
Lower reduction ops, introduce memory buffers
H
// Allocate buffers and transfer required data ...
gpu.launch (gridId.x i in [0, batchSize),
threadId.x t in [0, 3))
{
// Operations within the kernel are lowered to LIR
thresholds = loadThresholds(tree, tile)
featureIndices = loadFeatureIndices(tree, tile)
features = rows[i][featureIndices]
comparison = features < thresholds
tile = getChildTile(tree, tile, comparison)
}
// Transfer results back to CPU...
Lower to LLVM IR
I
LLVM IR
JIT to x86 + GPU
Memory Layout
GPU Array representation
GPU Sparse representation
GPU Reorg representation
J