Microsoft

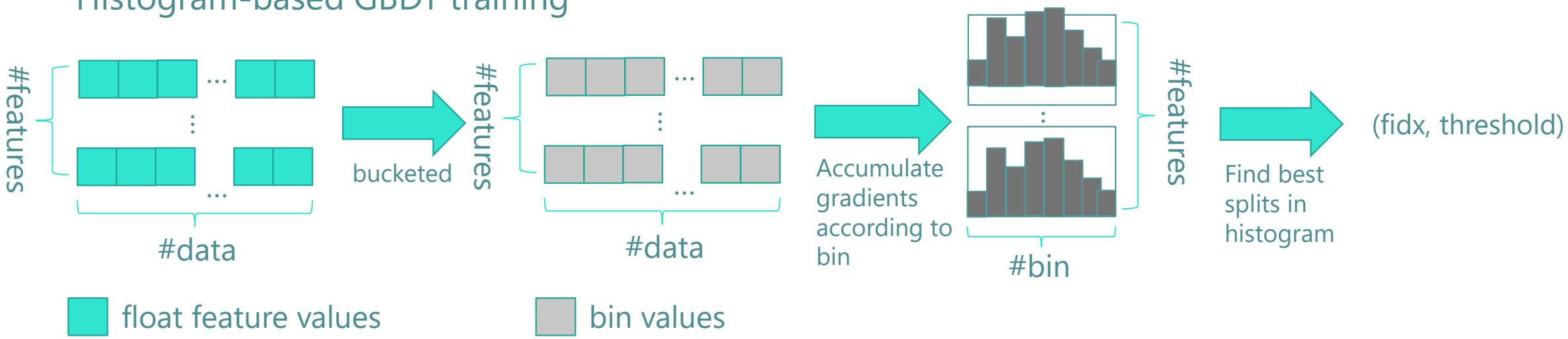
Quantized Training of Gradient Boosting Decision Trees

Yu Shi, Guolin Ke, Zhuoming Chen, Shuxin Zheng, Tie-Yan Liu

Microsoft Research

Intensive FP Operations in GBDT Training

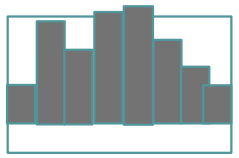
- Histogram-based GBDT training



- Problems in current GBDT training algorithm
  - Intensive arithmetic operations of 32/64-bit FP numbers, unable to exploit low precision computation
  - Large histograms causes low cache utility
  - Communication with 32/64-bit FP histograms when distributed training

Quantize Gradients in GBDT

- Training of GBDT is based on gradient statistics
  - Histogram construction



$\sum_i g_i$, $\sum_i h_i$ in each bin

  - Best split finding

$$\Delta\text{loss} = L_{\text{parent}} - L_{\text{left}} - L_{\text{right}} = \frac{G_{\text{left}}^2}{H_{\text{left}}} + \frac{G_{\text{right}}^2}{H_{\text{right}}} - \frac{G_{\text{parent}}^2}{H_{\text{parent}}}$$

$$G_{\text{leaf}} = \sum_{i \in \text{leaf}} g_i, \quad H_{\text{leaf}} = \sum_{i \in \text{leaf}} h_i$$

- Quantize 32-bit gradients g_i and h_i into low-bitwidth integers $\hat{g}_i$ and $\hat{h}_i$
- Accumulation of G_{leaf} and H_{leaf} can be done with lower cost (e.g., 8-bit, 16-bit, 32-bit)

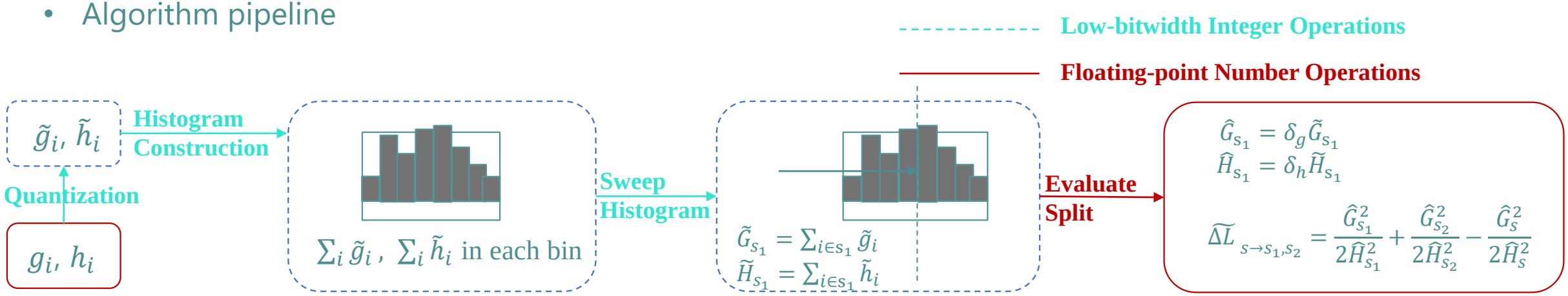
Quantized Training of GBDT

- Gradient Quantization: Equal-distance division of the gradient value range

$$\alpha = \frac{2 \cdot \max_j |g_j|}{B} \qquad \hat{g}_i \in \left\{-\frac{B}{2}, -\left(\frac{B}{2}-1\right), \ldots, -1, 0, 1, \ldots, \left(\frac{B}{2}-1\right), \frac{B}{2}\right\}$$

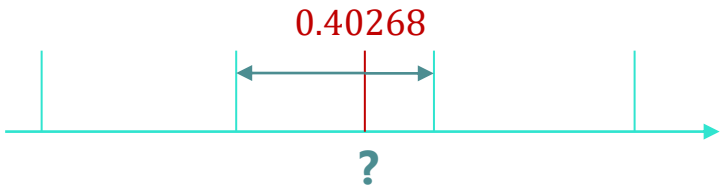
$$\beta = \frac{\max_j h_j}{B} \qquad \hat{h}_i \in \{0, 1, \ldots, (B-1), B\}$$

- Algorithm pipeline



Quantization Methods

- Quantization: Cast more values into fewer values



32-bit FP number
2-bit Integer

- Round-to-nearest

$$\mathrm{RN}(x) = \begin{cases} \lfloor x \rfloor, & x < \lfloor x \rfloor + \frac{1}{2} \\ \lceil x \rceil, & x \geq \lfloor x \rfloor + \frac{1}{2} \end{cases}$$

- Stochastic rounding

$$\mathrm{SR}(x) = \begin{cases} \lfloor x \rfloor, & \text{w.p.} \quad \lceil x \rceil - x \\ \lceil x \rceil, & \text{w.p.} \quad x - \lfloor x \rfloor \end{cases}$$

Analysis of Quantization Error

Theorem 5.3 For loss functions with constant hessian value $h > 0$, if Assumption 5.2 holds for the subset $\mathcal{D}_s$ in leaf s for some $\gamma_s > 0$, then with stochastic rounding and leaf-value refitting, for any $\epsilon > 0$, and $\delta > 0$, at least one of the following conclusions holds:

1. With any split of leaf s and its descendants, the resultant average of absolute values of prediction values by the tree in the current boosting iteration for data in $\mathcal{D}_s$ is no greater than ϵ/h.

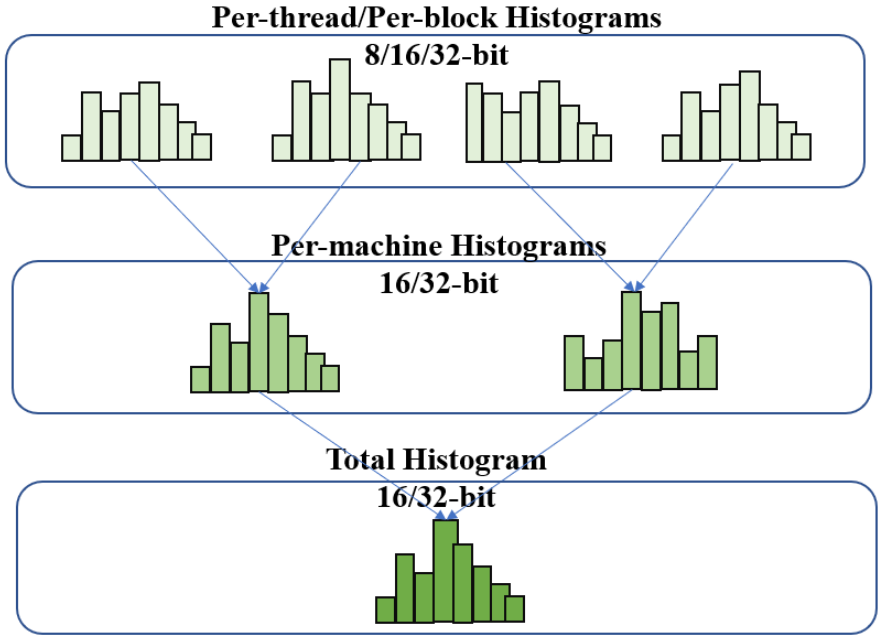
2. For any split $s \to s_1, s_2$ of leaf s, with a probability of at least $1 - \delta$,

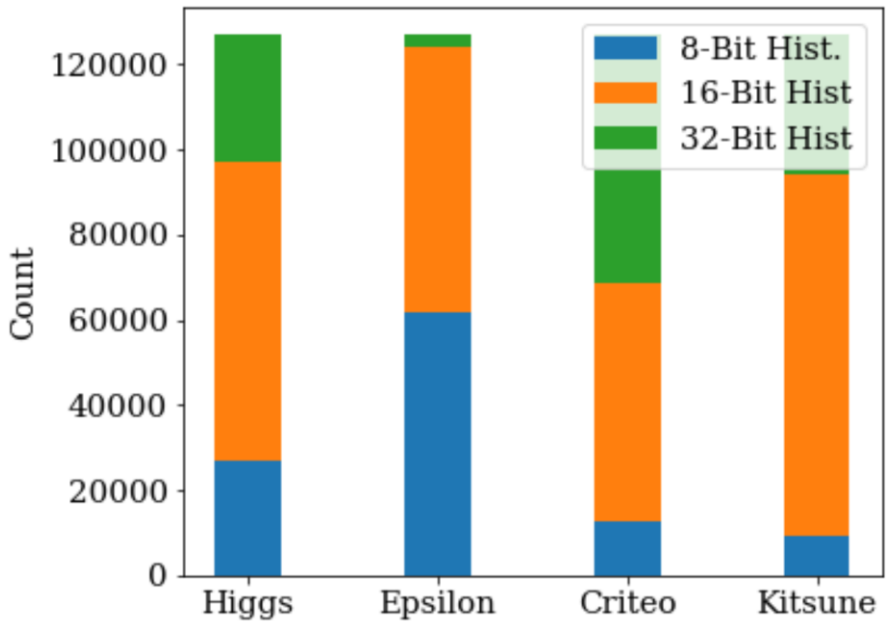
$$\frac{\left|\widetilde{\mathcal{G}}_{s\to s_1,s_2} - \mathcal{G}_{s\to s_1,s_2}\right|}{\mathcal{G}_s^*} \le \frac{\max\limits_{i\in[N]}|g_i|\sqrt{2\ln\frac{4}{\delta}}}{\gamma_s^2\epsilon\cdot 2^{B-1}}\left(\sqrt{\frac{1}{n_{s_1}}}+\sqrt{\frac{1}{n_{s_2}}}\right)+\frac{\left(\max\limits_{i\in[N]}|g_i|\right)^2\ln\frac{4}{\delta}}{\gamma_s^2\epsilon^2 n_s\cdot 4^{B-2}}. \tag{9}$$

- Either the split won't change prediction values much
- Or the split gain is well estimated with quantized gradients

System Implementation

- Hierarchical Histogram Buffers

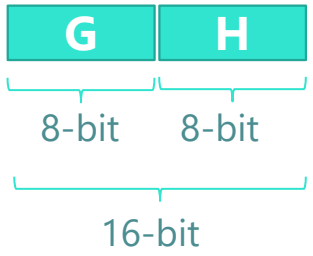




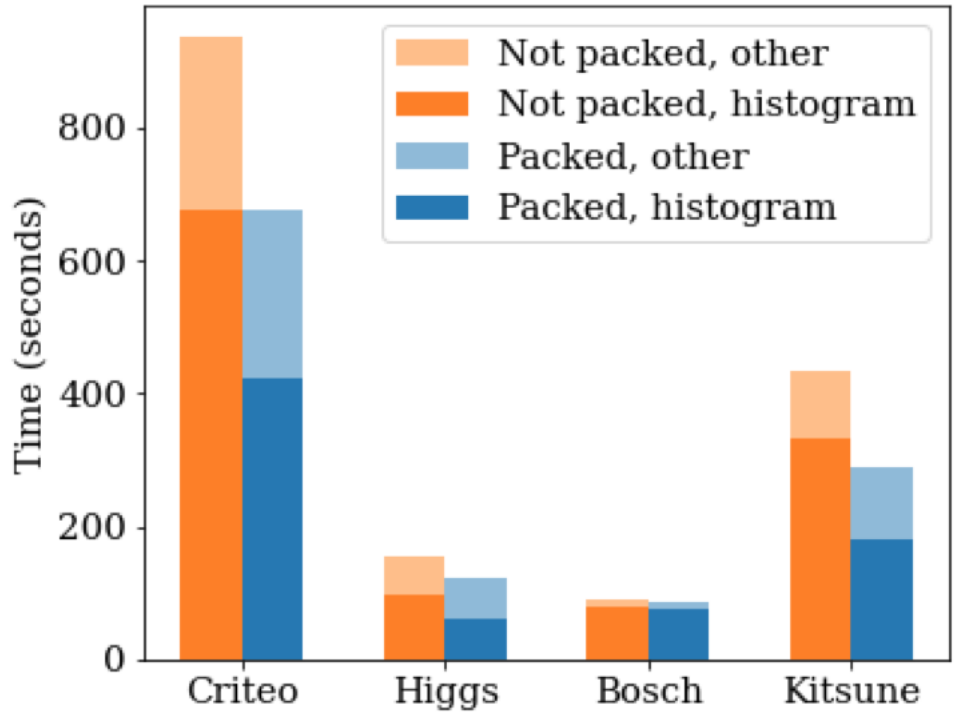
System Implementation

- Packing Gradient and Hessian

  Accumulate G and H in a single integer addition



  Compared with vectorization for FP addition
  - Vectorization without slowing down CPU frequer
  - Applicable on GPU

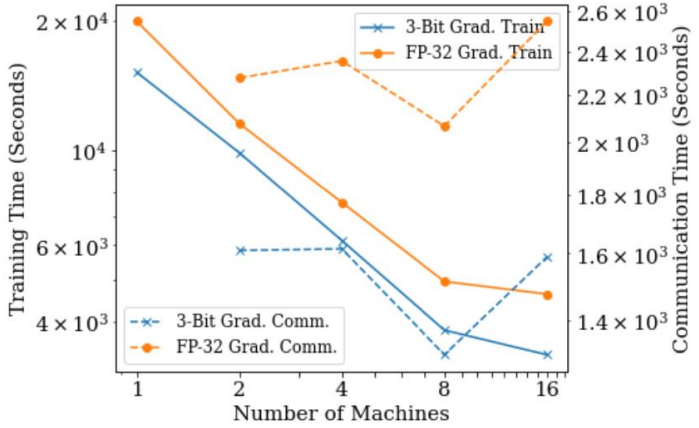


Accuracy of Quantized Training

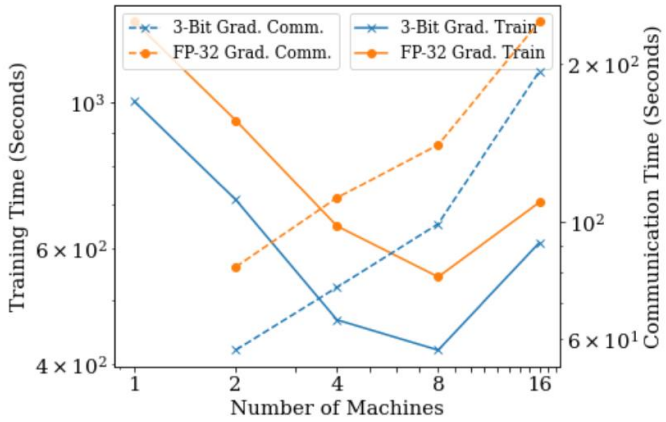
Bitwidth	Binary Classification					Regression	Ranking	
	Higgs↑	Epsilon↑	Kitsune↑	Criteo↑	Bosch↑	Year↓	Yahoo LTR↑	LETOR↑
32-bit	0.845694	0.950203	0.950561	0.803791	0.703101	8.956278	**0.793857**	0.524265
2-bit SR_{refit}	0.845587	0.949472	0.952703	0.803293	0.700322	8.953388	0.788579	0.519268
3-bit SR_{refit}	0.845725	0.949884	0.951309	0.803768	0.702756	**8.937374**	0.791077	0.522220
4-bit SR_{refit}	0.845507	0.950049	0.950911	0.803783	0.703315	8.942898	0.792664	0.523796
5-bit SR_{refit}	0.845706	0.950298	0.949229	0.803766	0.702971	8.948542	0.793166	**0.524673**
2-bit $SR_{no\ refit}$	**0.846713**	0.944509	0.952974	0.803750	0.700900	9.112302	0.764862	0.486193
3-bit $SR_{no\ refit}$	0.846040	0.949593	0.951385	**0.803922**	0.702501	8.990034	0.780041	0.507689
4-bit $SR_{no\ refit}$	0.845816	0.950127	0.951197	0.803812	0.703327	8.955256	0.787575	0.515767
5-bit $SR_{no\ refit}$	0.845842	0.950275	0.949794	0.803790	0.703226	8.952768	0.791631	0.520900
2-bit RN_{refit}	0.795991	0.889149	0.962201	0.779906	0.686617	9.429014	0.765103	0.454894
3-bit RN_{refit}	0.830506	0.944329	**0.966606**	0.782732	0.688899	9.062854	0.772364	0.476726
4-bit RN_{refit}	0.840747	0.949946	0.961938	0.795803	0.691469	8.968694	0.777347	0.487256
5-bit RN_{refit}	0.843820	**0.950457**	0.962427	0.802438	0.698954	8.952418	0.784333	0.494951
2-bit $RN_{no\ refit}$	0.836683	0.925220	0.946016	0.768338	0.704445	10.685840	0.632058	0.203732
3-bit $RN_{no\ refit}$	0.843482	0.946850	0.940961	0.791709	**0.708724**	9.377560	0.732487	0.350127
4-bit $RN_{no\ refit}$	0.845788	0.949676	0.949228	0.802689	0.703718	8.969828	0.765432	0.437317
5-bit $RN_{no\ refit}$	0.845765	0.950307	0.952420	0.803645	0.698419	8.965400	0.782608	0.485752

Speedup of Quantized Training

	Algorithm	Bosch	Criteo	Epsilon	Higgs	Kitsune	Year	Yahoo LTR	LETOR
GPU total time	XGBoost	73	373	125	28	195	**15**	41	48
	CatBoost	21	199	100	59	81	32	58	N/A
	LightGBM+	22	101	86	28	77	24	30	41
	LightGBM+ 2-bit	13	62	**38**	**24**	**37**	18	23	**33**
	LightGBM+ 3-bit	13	60	39	24	39	18	**24**	34
	LightGBM+ 4-bit	13	**59**	39	**24**	40	**17**	26	**33**
	LightGBM+ 5-bit	**12**	**59**	41	**24**	40	**17**	25	34
CPU total time	XGBoost	326	1243	2697	201	606	62	213	155
	CatBoost	2829	11880	1659	1607	2023	130	761	1283
	LightGBM	109	863	846	149	454	23	136	166
	LightGBM 2-bit	**84**	764	808	136	309	24	**95**	125
	LightGBM 3-bit	87	730	**775**	**125**	**289**	23	101	**123**
	LightGBM 4-bit	87	706	**775**	131	291	23	102	129
	LightGBM 5-bit	**84**	**678**	776	127	338	**22**	105	128
GPU Histogram time	LightGBM+	17	70	46	11	54	9	11	17
	LightGBM+ 2-bit	**8**	21	**12**	**4**	**16**	**4**	**8**	**10**
CPU Histogram time	LightGBM	98	629	737	94	339	12	108	109
	LightGBM 2-bit	**72**	458	708	68	203	10	**67**	**68**



(a) Epsilon-8M



(b) Criteo

Thank You