Yandex

Yandex

Using GPUs to Accelerate Learning to Rank

Alexander Shchekalev
Senior developer

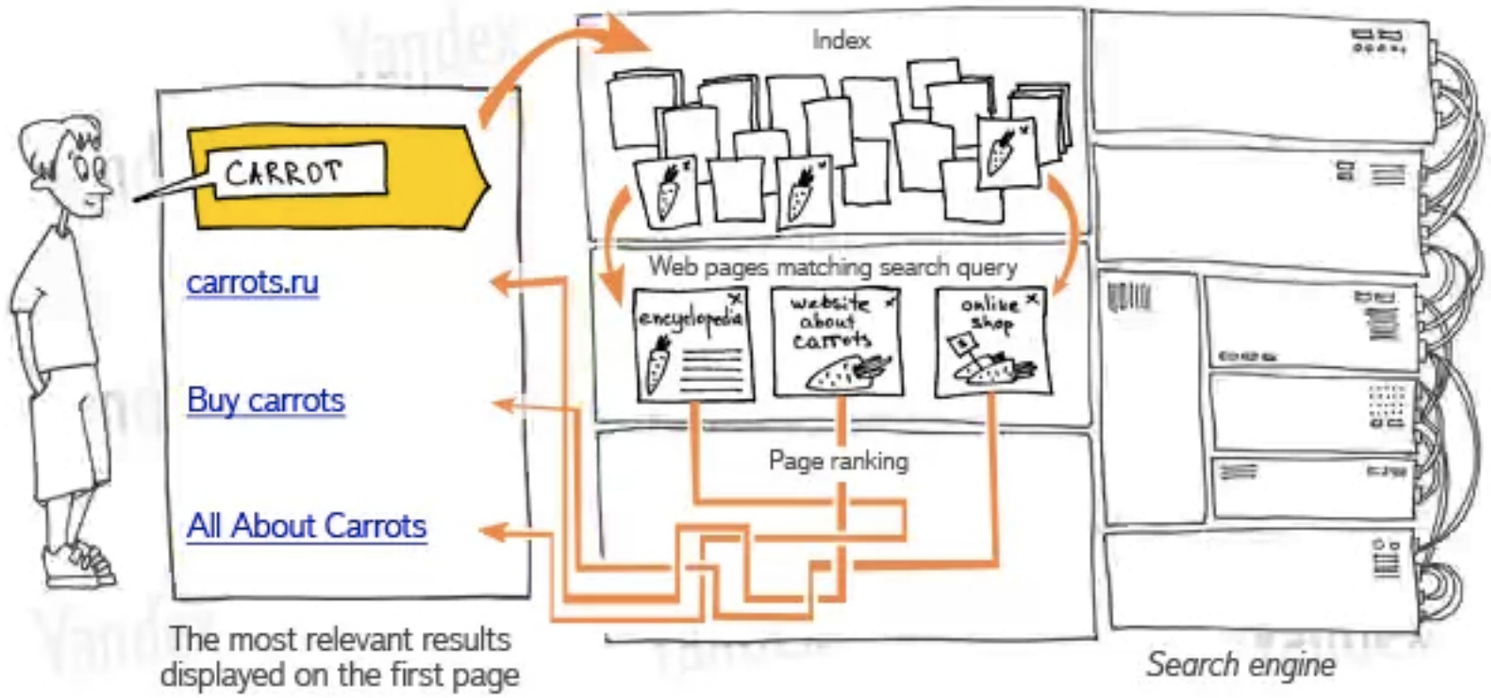
Contents

- Task of learning to rank
- Common approaches in solving the task
- Decision tree boosting
- MatrixNet and its GPU implementation
- Performance
- Applications

Yandex’s search engine

- Search engine provides answers to user’s queries
- Millions of documents match any search query
- Ranking defines the order of search results to provide a user with the most relevant ones

Yandex’s search engine



- Millions of documents match any search query
- Ranking defines the order of search results to provide a user with the most relevant ones

Yandex's machine learning

- Prerequisites:
  - Set of search queries $Q = \{q_1, ..., q_n\}$
  - Set of documents corresponding to each query $q \to \{d_1, d_2, ...\}$
  - Relevance judgements for each query-document pair l_{qd}
- Each query-document pair is represented in the vector space of features (number of links, word matching etc.)
- Ranking quality is estimated by an evaluation measure (MAP, DCG etc.)

Solving learning to rank problem

- There are three common approaches:
  - Listwise
  - Pairwise
  - Pointwise
- Listwise approach optimizes the evaluation measure
- Pairwise approach uses a classification problem and optimizes the objective functions on the pairs of documents

Pointwise approach (regression)

- Pointwise approach optimizes relevance predictions

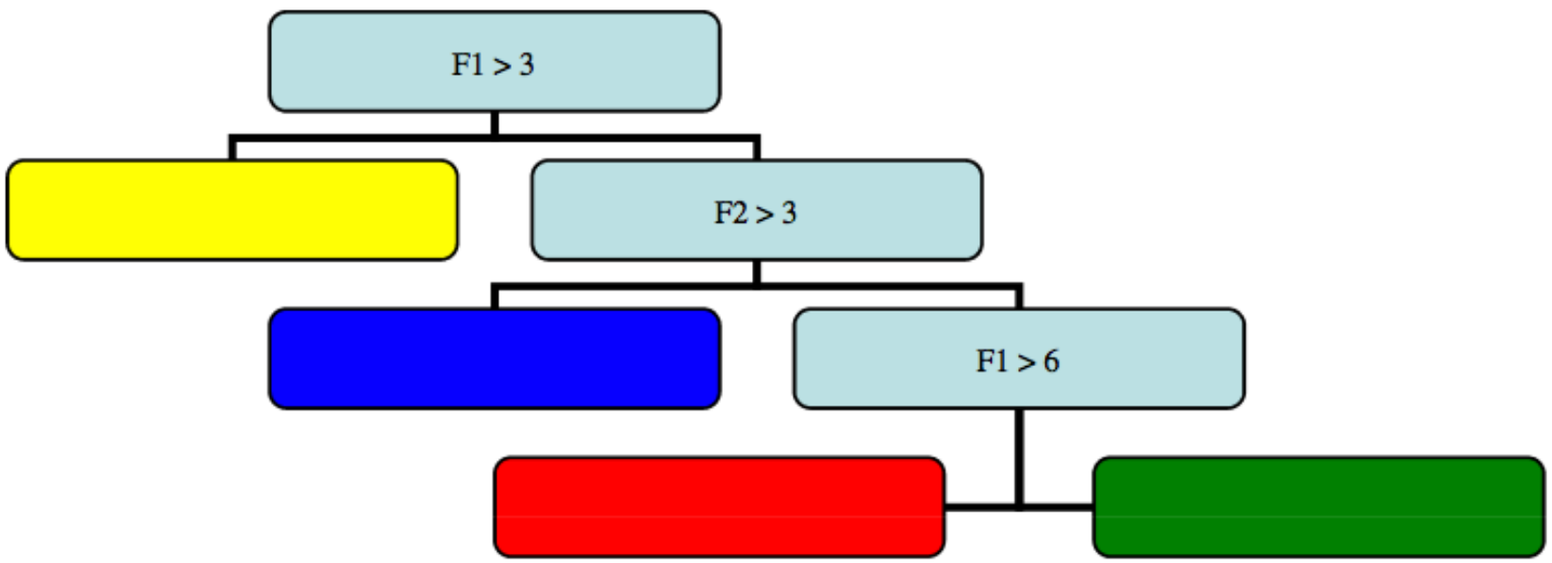
$$\arg\min_{fr \in F} \frac{1}{n} \sum_{(q,d)} L(fr(q,d), rel(q,d))$$

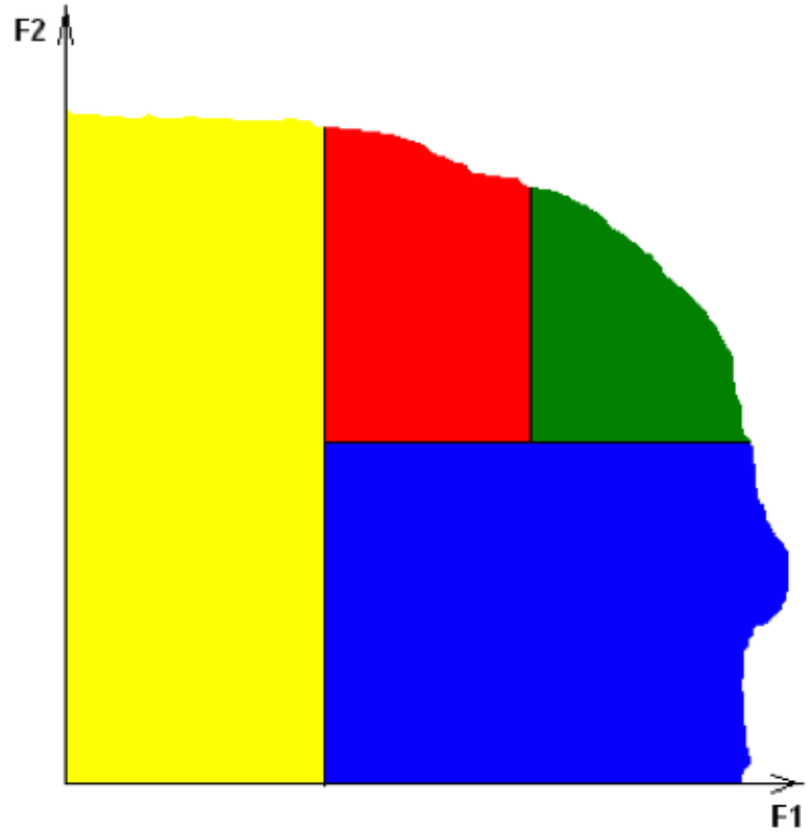
- $L(fr(q,d), rel(q,d))$ – loss function, F – set of possible ranking functions

Boosting

- Ensemble method builds a strong learner as a linear combination of weak learners
- Trade-off between the quality of model and the time of the learning process
- Choice of decision trees as weak learners shows the best results for the learning-to-rank problem

Decision trees





MatrixNet

- Method is based on the gradient boosting of decision trees
- Each decision tree is binary and consists of 6 splits
- Sorted values of each feature are split into 32 equal size bins and bin min/max values are used as possible splits

MatrixNet

- GPU based implementation calculates 32-bin histograms to evaluate each available split of one feature
- Histogram construction makes extensive use of very fast shared memory
- Gradient values are stored in registers and loaded from global memory only once for all splits of feature

Histogram calculation

- Single thread block uses all available shared memory
- Block size equals 384 threads for Fermi and Kepler architectures
- Simple layout in case of single feature per block

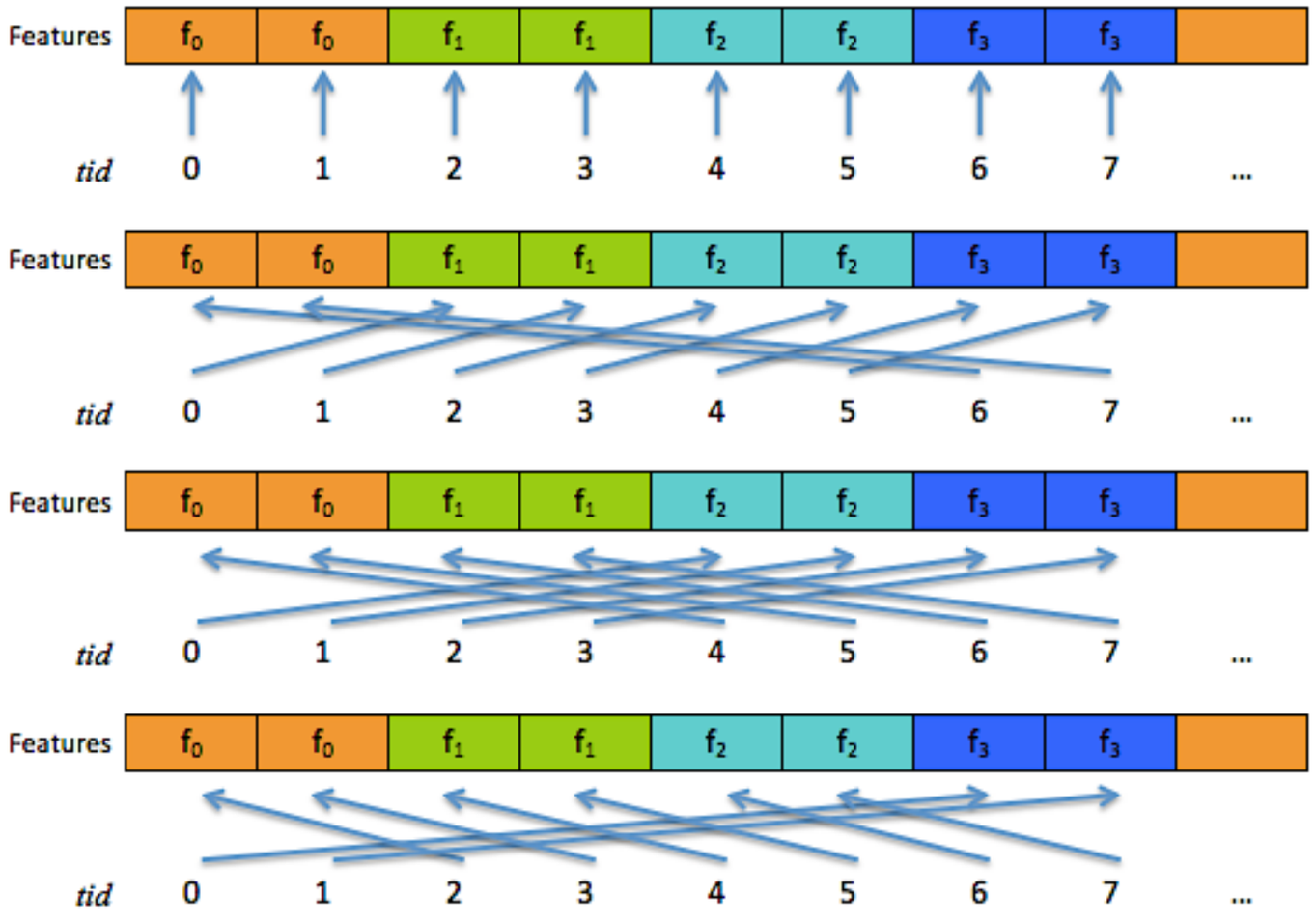
$$tid + b_{tid} * 384$$

Histogram calculation

- Each warp builds partial histogram using 1024 elements with the following layout

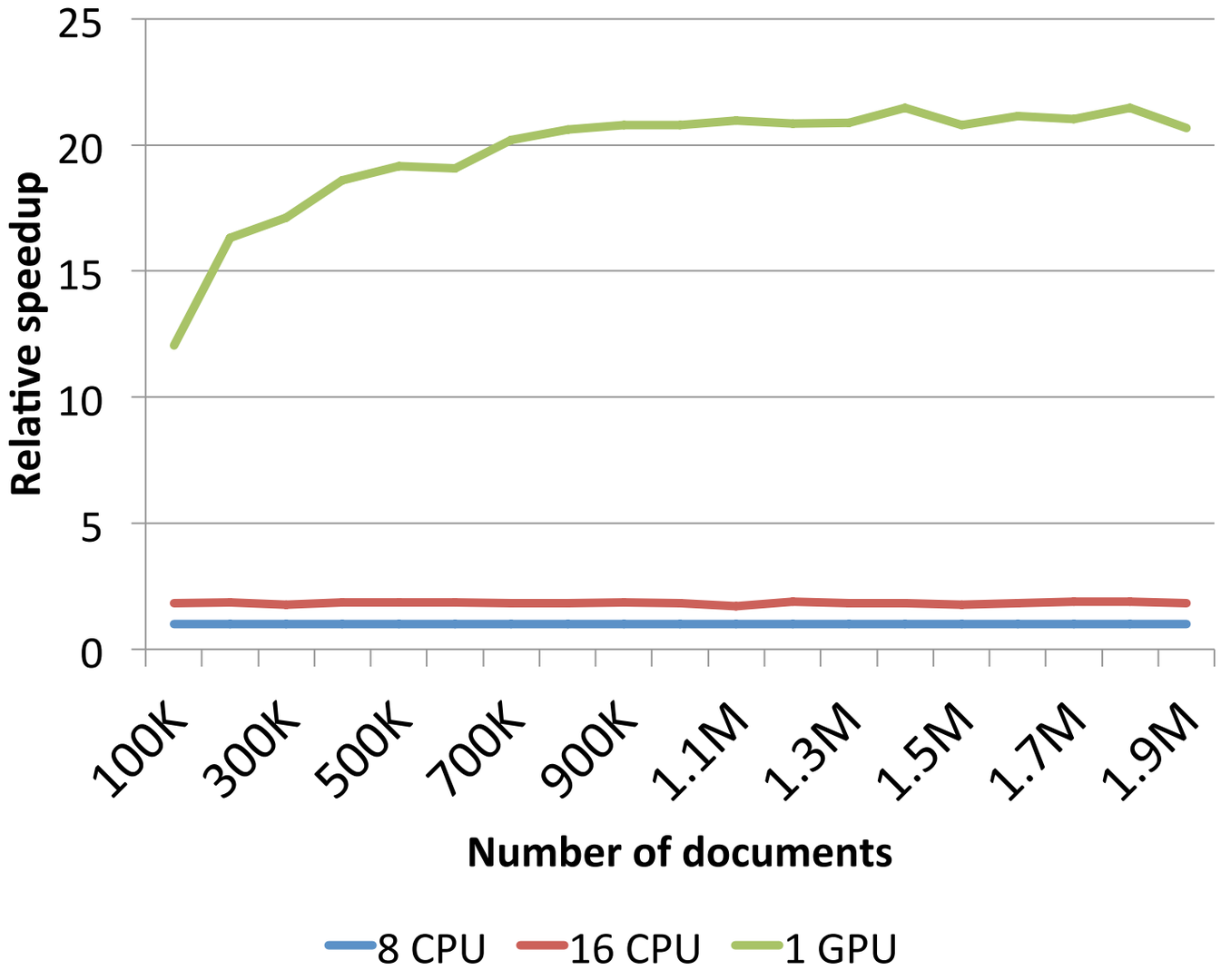
$$f_k = (tid \ \& \ 7 + k * 2) + (tid \ \& \ 1) + (tid \ \& \ 24)$$

$$f_k + b_{tid} * 32$$



Performance comparison

- Hardware
  - CPU Intel Xeon E5-2650
  - GPU NVIDIA Tesla C2075
- Data
  - Common data set used for training production ranking models
  - Includes more than 700 features
- More than 20 times speed up



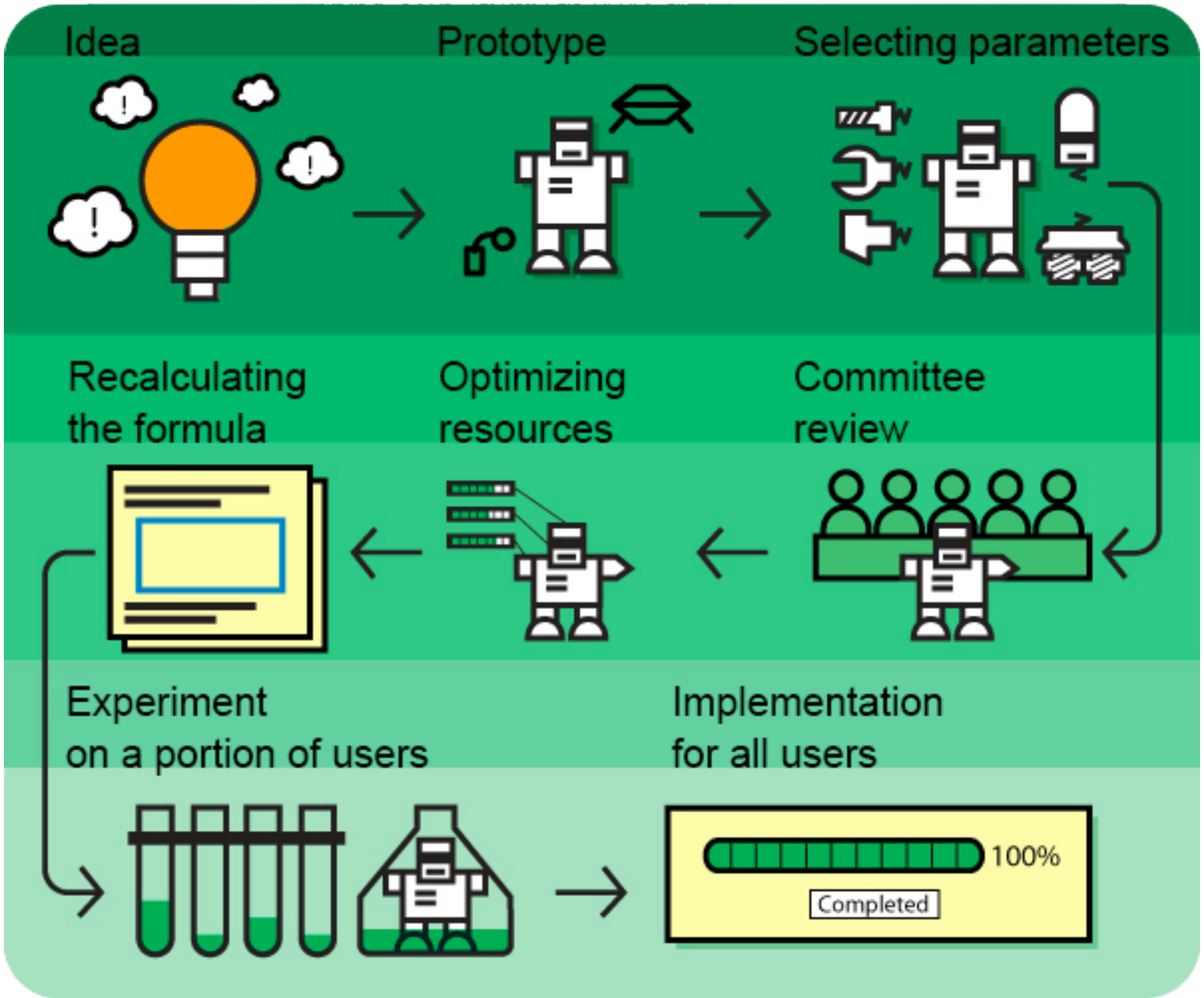
Application: ranking functions

- Pairwise target function is used to train production models
- CPU profile
  - Hardware: 50 double CPU machines
  - Time: 25 hours
- GPU profile
  - Hardware: 1 machine with 4 Tesla M2090
  - Time: 8 hours
- Workload of 1000 models per month is handled by 12 machines

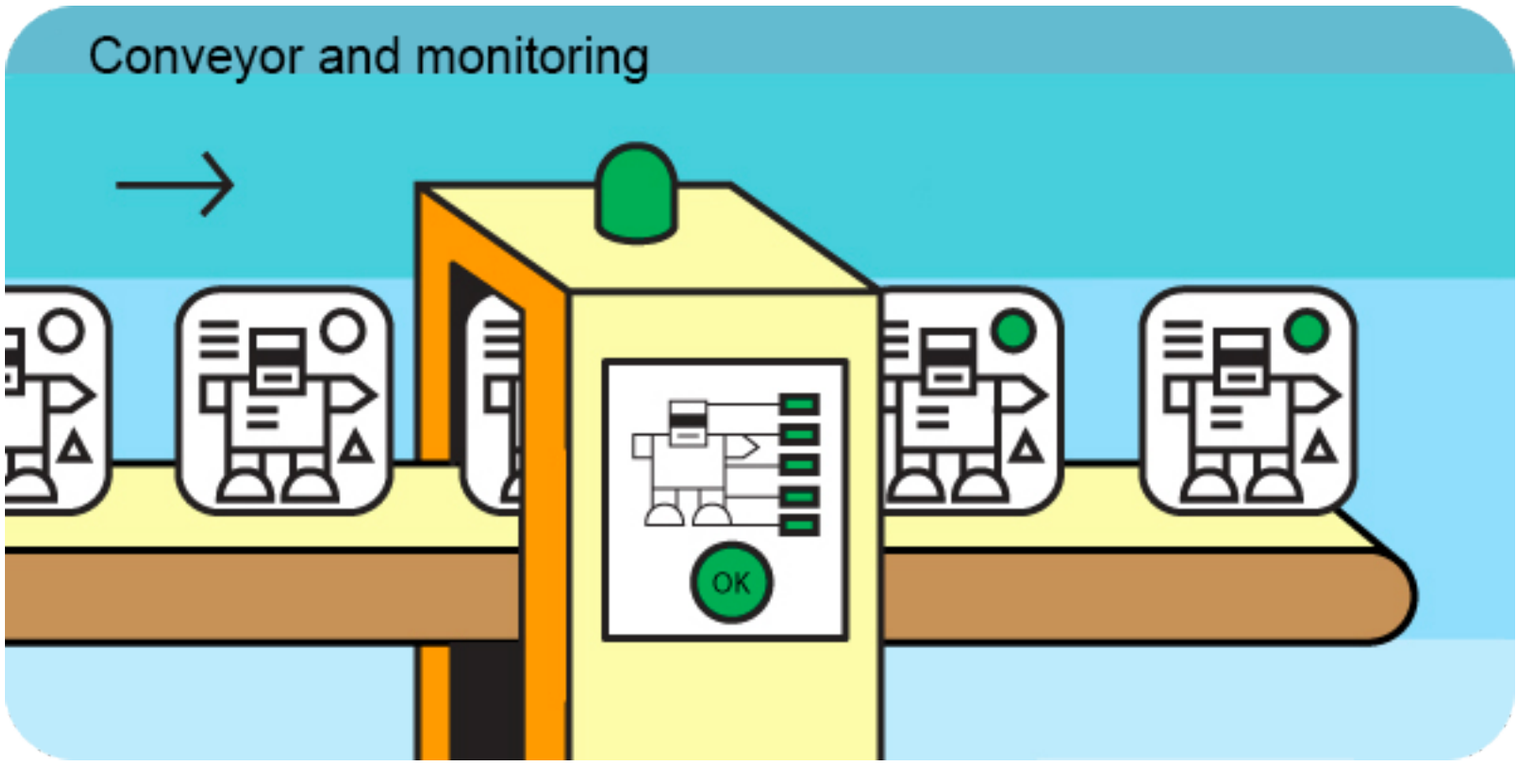
Application: evaluating features

- New ranking features help better understand the user
- Results of 250 pairs of models are compared by cross validation
- CPU profile
  - Hardware: 500 CPU machines
  - Time: 1 hour
- GPU profile
  - Hardware: 6 machines with 4 GPUs
  - Time: 1 hour
- Almost 7000 evaluations are handled by the cluster with 40 machines

Friendly Machine Learning



Friendly Machine Learning



Yandex Data Factory

- Public version of FML – Yandex Data Factory
- Full cycle machine learning pipeline – from research to production

Yandex

Alexander Shchekalev

Senior developer

sancho@yandex-team.ru

Thank you!