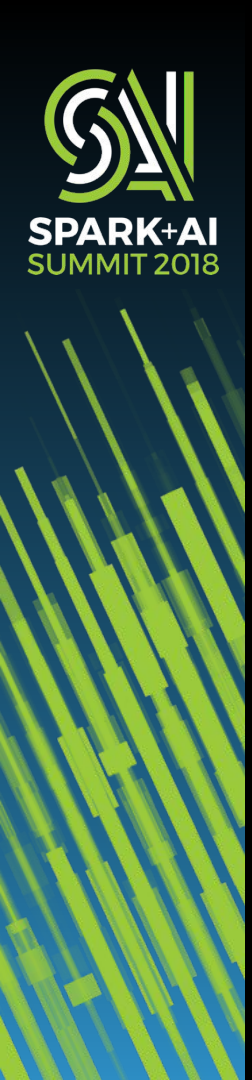
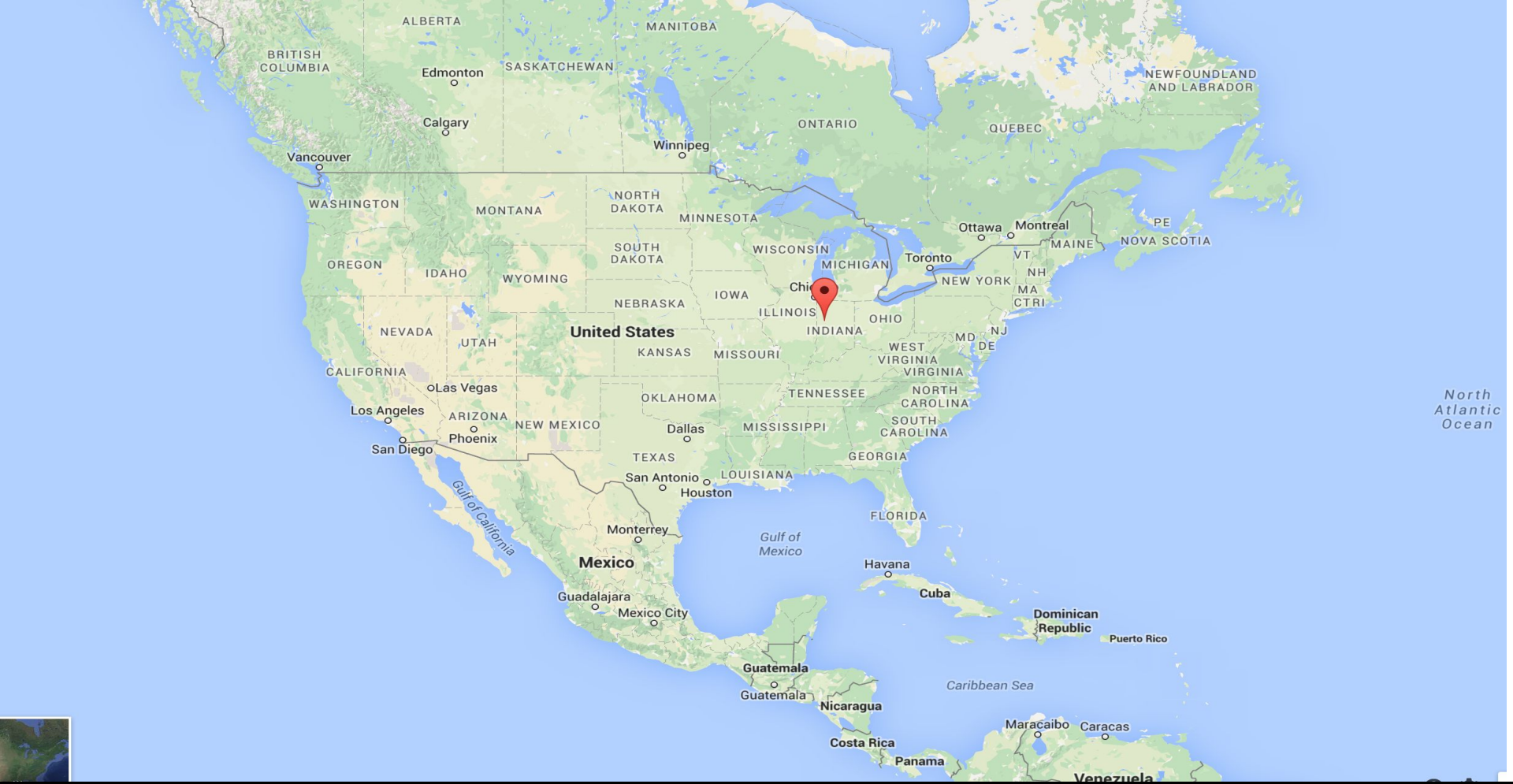


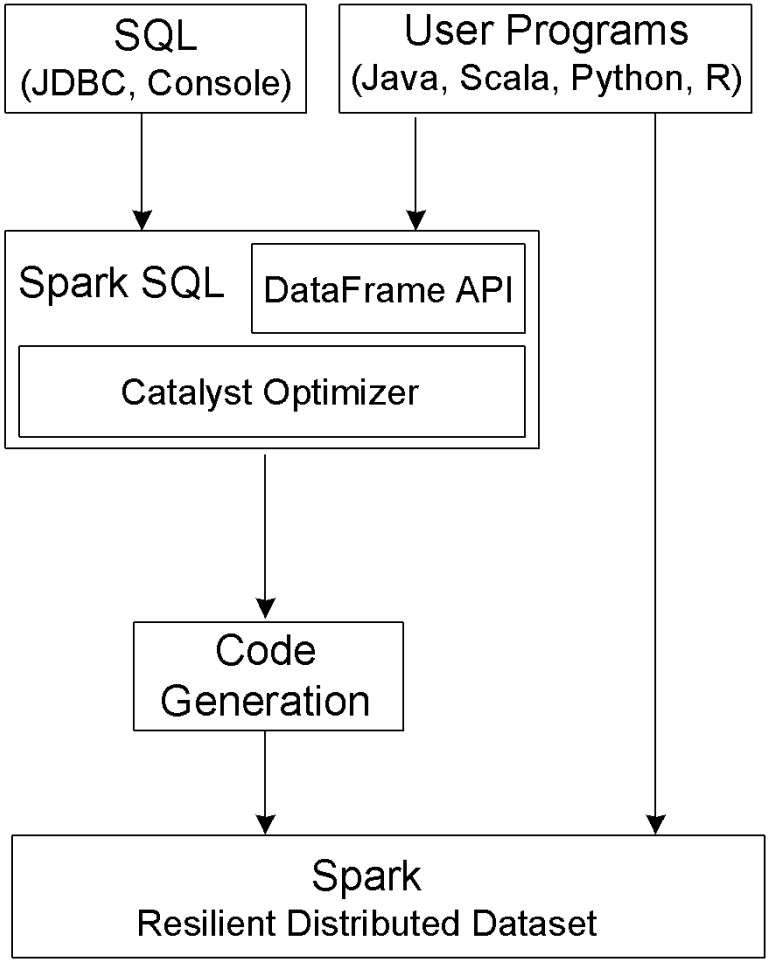
Flare and TensorFlare: Native Compilation for Spark and TensorFlow Pipelines

Gregory Essertel, Purdue University
Tiark Rompf, Purdue University

#Res5SAIS


ALBERTA
MANITOBA
BRITISH COLUMBIA
Edmonton
SASKATCHEWAN
NEWFOUNDLAND AND LABRADOR
Calgary
ONTARIO
QUEBEC
Winnipeg
Vancouver
WASHINGTON
MONTANA
NORTH DAKOTA
MINNESOTA
Ottawa
Montreal
PE
NOVA SCOTIA
SOUTH DAKOTA
WISCONSIN
MAINE
OREGON
IDAHO
WYOMING
MICHIGAN
Toronto
VT
NEW YORK
NH
MA
CTRI
NEBRASKA
IOWA
ILLINOIS
OHIO
NEVADA
UTAH
United States
INDIANA
MD
NJ
DE
KANSAS
MISSOURI
WEST VIRGINIA
VIRGINIA
CALIFORNIA
Las Vegas
OKLAHOMA
TENNESSEE
NORTH CAROLINA
Los Angeles
ARIZONA
NEW MEXICO
Dallas
MISSISSIPPI
SOUTH CAROLINA
Phoenix
San Diego
TEXAS
GEORGIA
San Antonio
LOUISIANA
Houston
Gulf of California
FLORIDA
Monterrey
Gulf of Mexico
Mexico
Havana
Guadalajara
Cuba
Mexico City
Dominican Republic
Puerto Rico
Guatemala
Guatemala
Caribbean Sea
Nicaragua
Maracaibo
Caracas
Costa Rica
Panama
Venezuela
North Atlantic Ocean

APACHE
Spark
TM

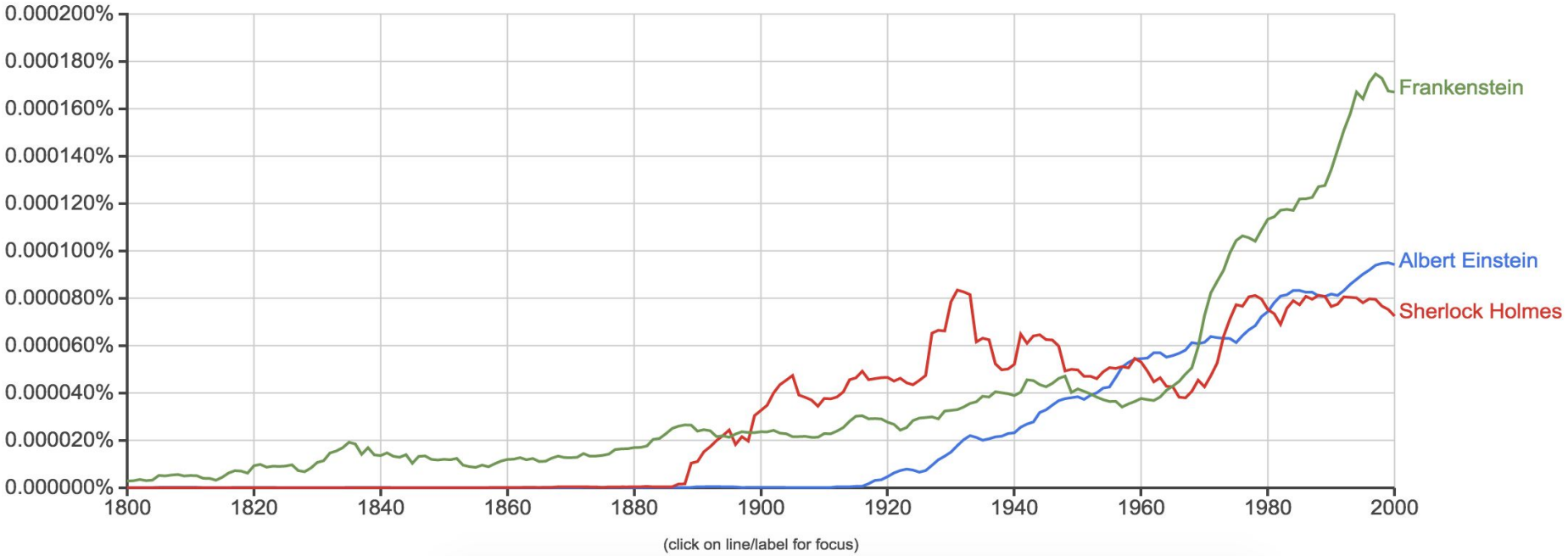

SQL
(JDBC, Console)
User Programs
(Java, Scala, Python, R)
Spark SQL
DataFrame API
Catalyst Optimizer
Code
Generation
Spark
Resilient Distributed Dataset

How Fast Is Spark?

Google Books Ngram Viewer

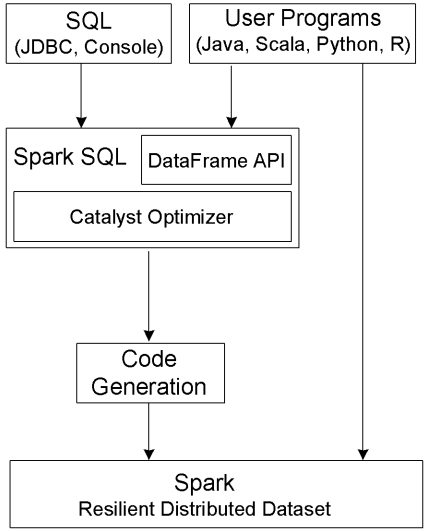
Graph these comma-separated phrases: Albert Einstein,Sherlock Holmes,Frankenstein case-insensitive

between 1800 and 2000 from the corpus English with smoothing of 3. Search lots of books

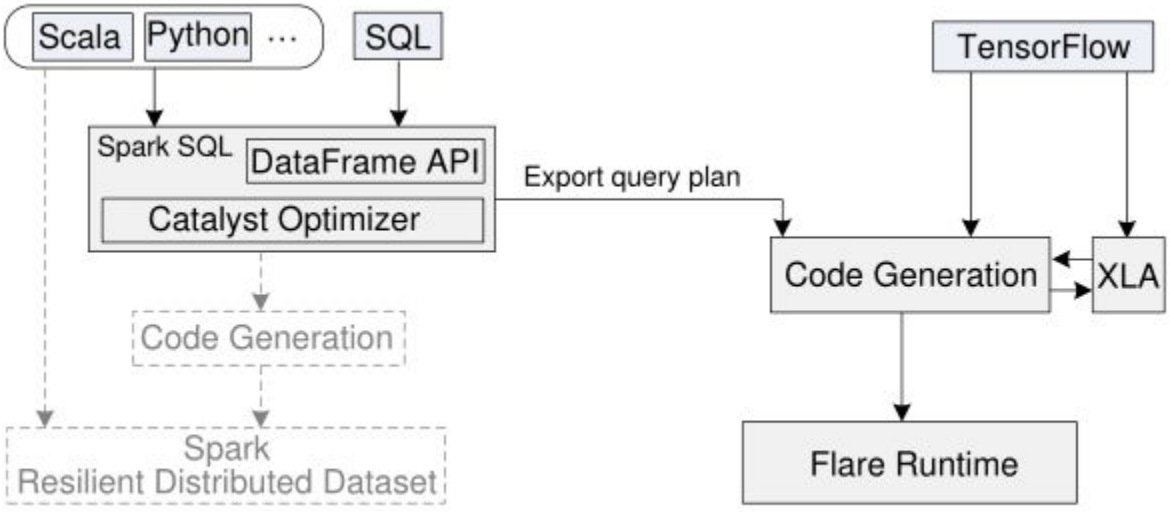


Demo

Spark Architecture



Flare: a New Back-end for Spark

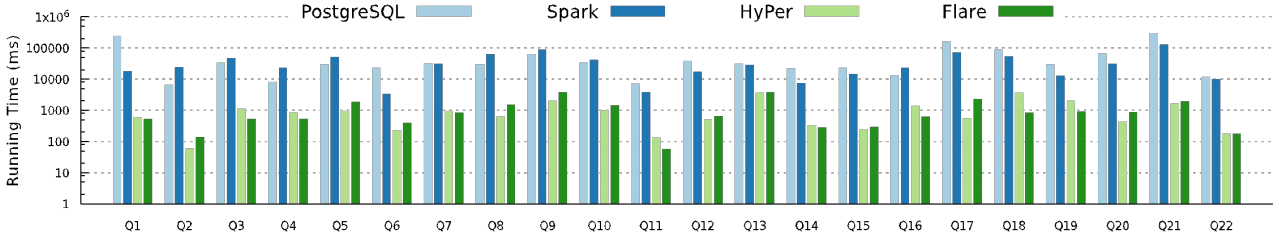


Results

Single-Core Running Time: TPCH

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Postgres	241404	6649	33721	7936	30043	23358	32501	29759	64224	33145	7093
Spark	18219	23741	47816	22630	51731	3383	31770	63823	88861	42216	3857
Hyper	603	59	1126	842	941	232	943	616	1984	967	131
Flare	529	139	536	520	747	365	828	1533	3131	1795	56

	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22
Postgres	37880	31242	22058	23133	13232	155449	90949	29452	65541	299178	11703
Spark	17233	28489	7403	14542	23371	70944	53932	13085	31226	173778	10030
Hyper	501	3625	330	253	1399	563	3703	1980	434	1626	180
Flare	654	3715	260	303	620	2338	825	908	870	1963	177

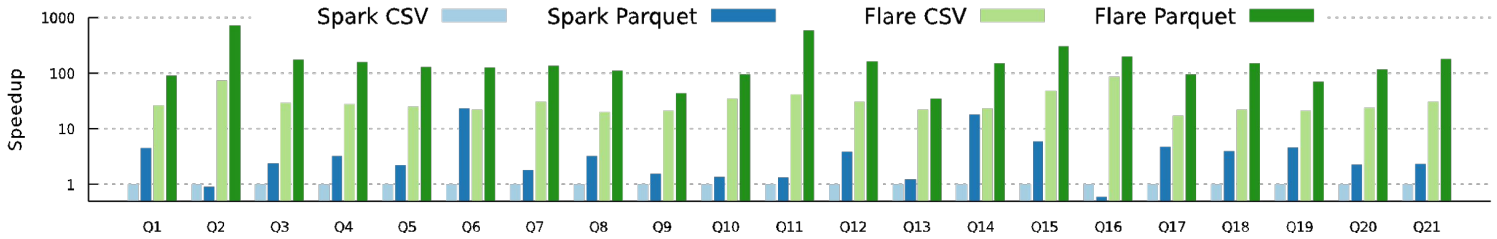


Absolute running time in milliseconds (ms) for Postgres, Spark, HyPer and Flare in SF10

Apache Parquet Format

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Spark CSV	16762	12244	21730	19836	19316	12278	24484	17726	30050	29533	5224
Spark Parquet	3728	13520	9099	6083	8706	535	13555	5512	19413	21822	3926
Flare CSV	641	168	757	698	758	568	788	875	1417	854	128
Flare Parquet	187	17	125	127	151	99	183	160	698	309	9

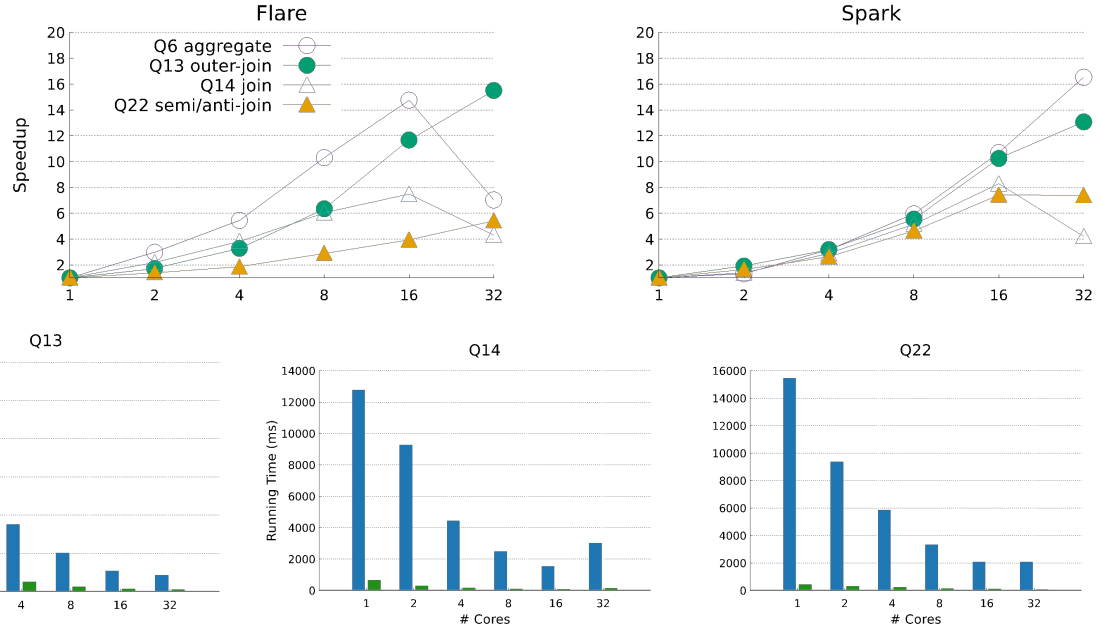
	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22
Spark CSV	21688	8554	12962	26721	12941	24690	27012	12409	19369	57330	7050
Spark Parquet	5570	7034	719	4506	21834	5176	6757	2681	8562	25089	5295
Flare CSV	701	388	573	551	150	1426	1229	605	792	1868	178
Flare Parquet	133	246	86	88	66	264	181	178	165	324	22



What about parallelism?

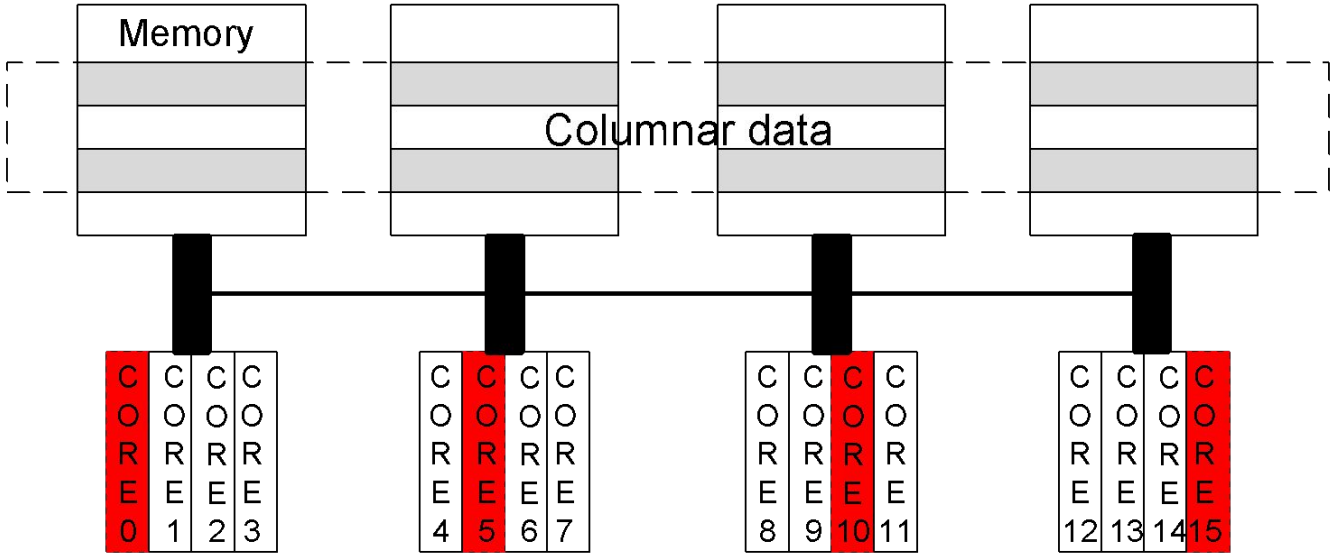
Parallel Scaling Experiment

Scaling-up Flare and Spark SQL in SF20

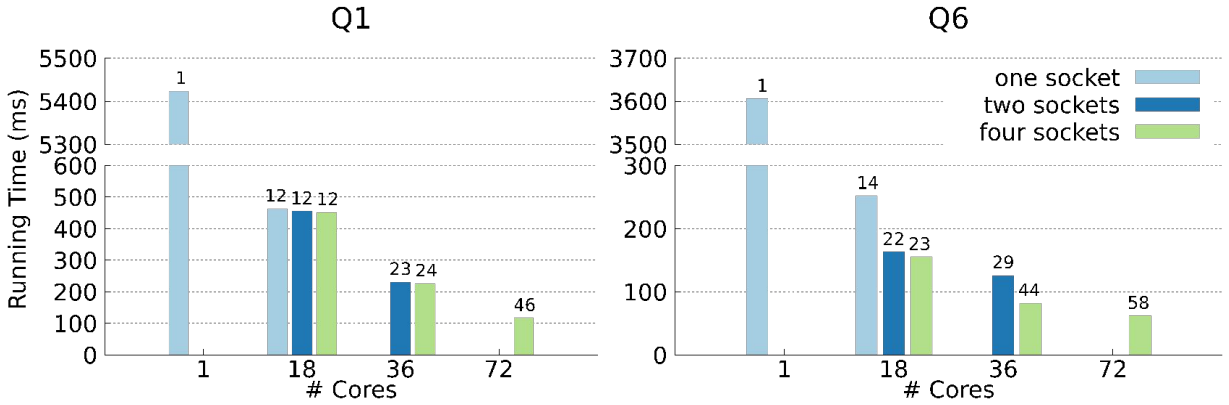


Hardware: Single NUMA machine with 4 sockets, 18 Xeon E5-4657L cores per socket, and 256GB RAM per socket (1 TB total).

NUMA Optimization



NUMA Optimization



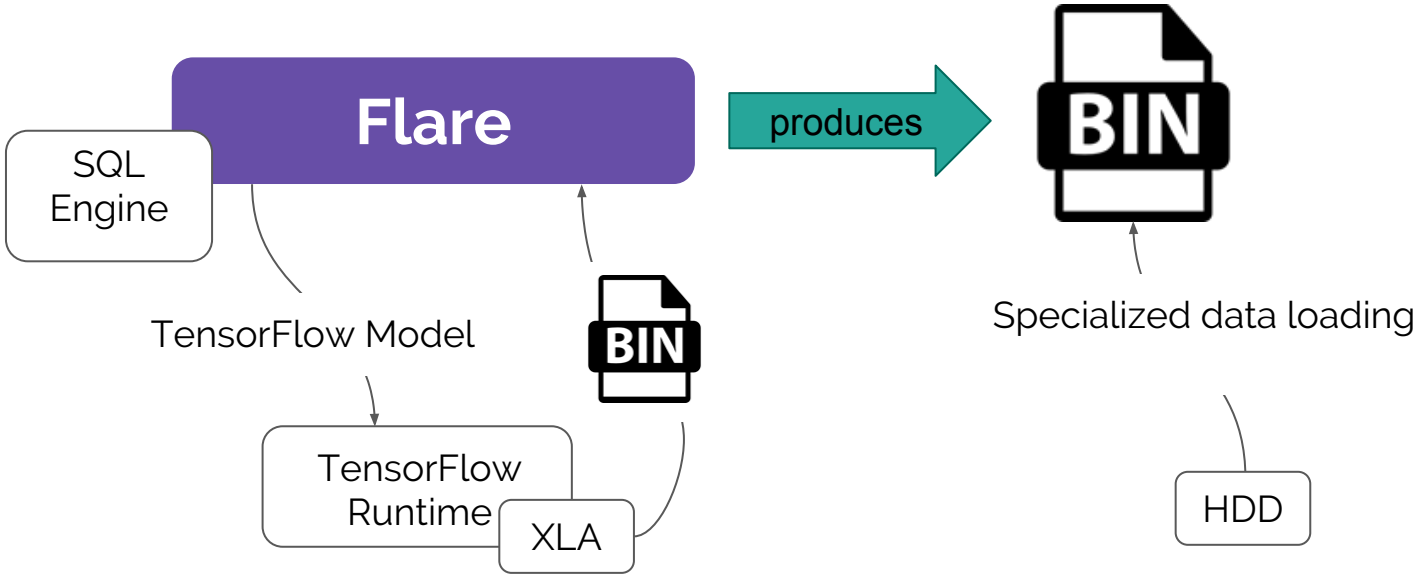
Scaling-up Flare for SF100 with NUMA optimizations on different configurations: threads pinned to one, two and four sockets

Hardware: Single NUMA machine with 4 sockets, 18 Xeon E5-4657L cores per socket, and 256GB RAM per socket (1 TB total).

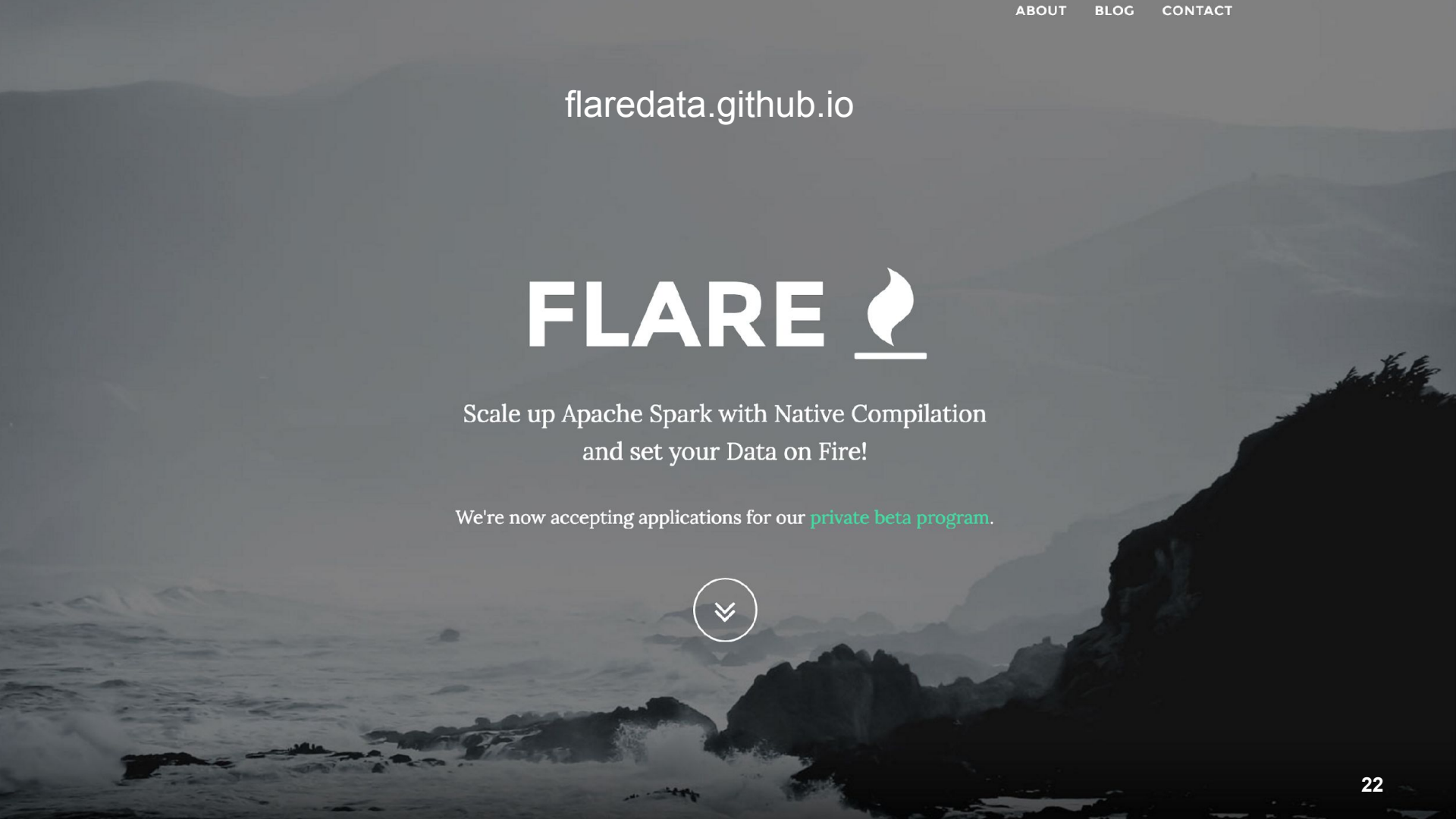
Heterogeneous Workloads: UDFs and ML Kernels

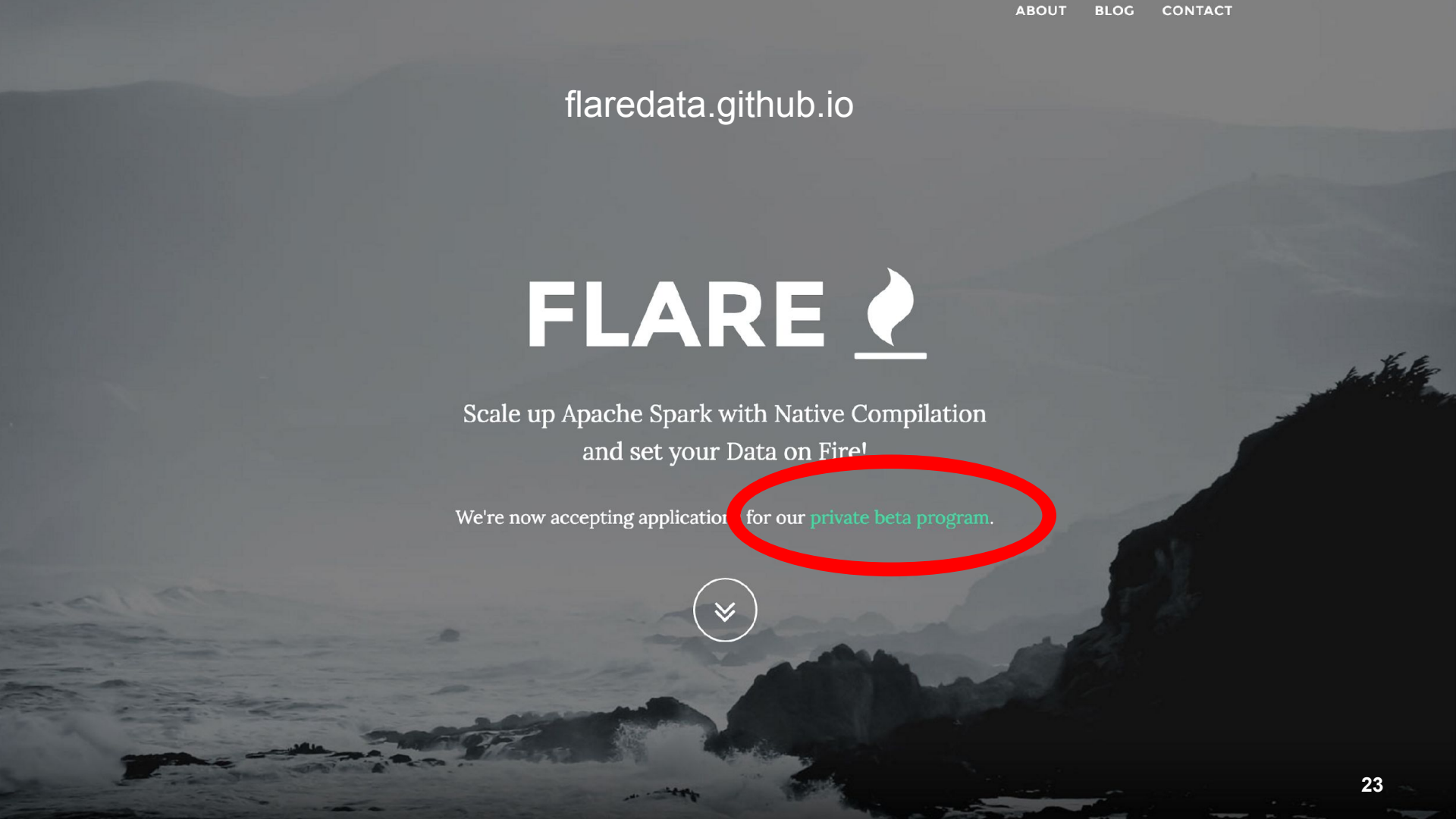
TensorFlow -> TensorFlare

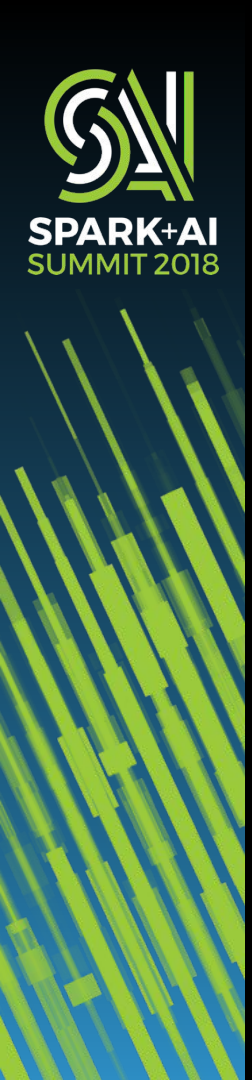
TensorFlare architecture



Demo


ABOUT
BLOG
CONTACT
flaredata.github.io
FLARE
Scale up Apache Spark with Native Compilation
and set your Data on Fire!
We're now accepting applications for our private beta program.


ABOUT
BLOG
CONTACT
flaredata.github.io
FLARE
Scale up Apache Spark with Native Compilation
and set your Data on Fire!
We're now accepting applications for our private beta program.



Thank You!

Web: flaredata.github.io

Twitter: @flaredata