2017 BENCHMARKING WORKING GROUP SURVEY *SUPPLEMENTAL*

February 2018

Overview/Methodology

- This report presents selected findings from the 2017 Benchmarking Working Group Survey
- The primary objective of the research was to help prioritize the efforts of the benchmarking team in 2018.
- The study was conducted online from 12/20/2017 to 1/31/2018 via a self-administered survey.
- The survey was fielded worldwide in English.
- The survey link was distributed by the Node.js Foundation through a number of channels including email, Twitter and by the Benchmarking Working Group and other Working Groups.
- A total of **294** individuals responded to at least some questions in the survey.*
- The data is available to the community in two ways
    1. The entire .xls dataset is available on GitHub
    2. A PDF summary along with this supplemental that visualizes responses to seven open text questions that SurveyMonkey does not include in their PDF export

*** Due to skip logic or missed questions, the number of responses for individual questions may be considerably lower**

Q5: What are the top 5 Node modules that you use most often?

1 Bluebird Restify Webpack React Lodash Expressjs Express Babel Request Koa Async Path Hapi

2 Mocha Graphql Webpack Async Babel Mongodb Express Path Lodash Sequelize Mongoose Moment Request Winston Koa Bunyan React

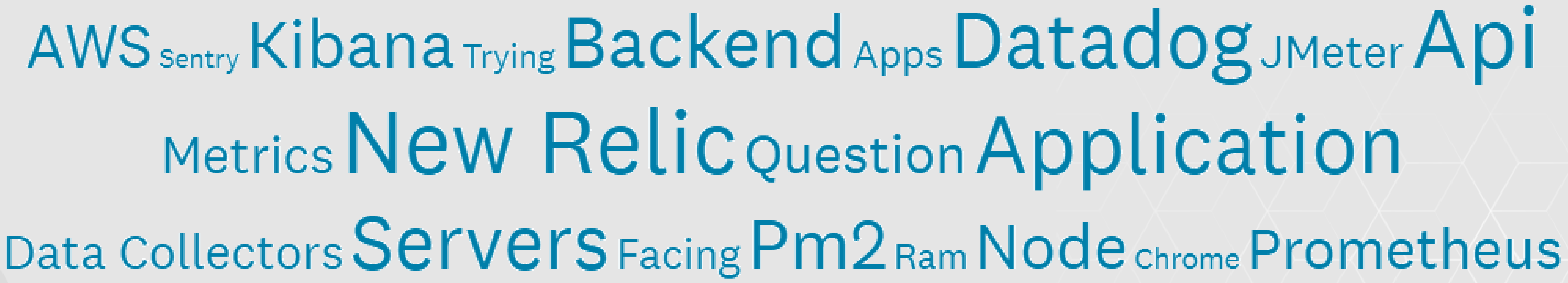
3 Hapi Nconf Bluebird Http-server Eslint Graphql Babel Uglify React Stream Webpack Mysql Lodash Browserify Express Sequelize Mongoose Typescript Mocha Glob Request Helmet Moment Mongodb Path Winston Chai Async

4 Winston Mongoose Request Crypto Angular Cheerio Knex React Lodash Events Async Body-parser Mocha Bluebird Express Chai Eslint Mysql Babel Bunyan Typescript Cluster Moment Jsonwebtoken Path Yargs Webpack UUID

5 Typescript Joi Async Util Eslint Restify React Redis Babel Mysql Mongoose Axios Lodash Prettier Mocha Request Moment Sequelize Winston Bluebird Webpack Knex Passport PM2

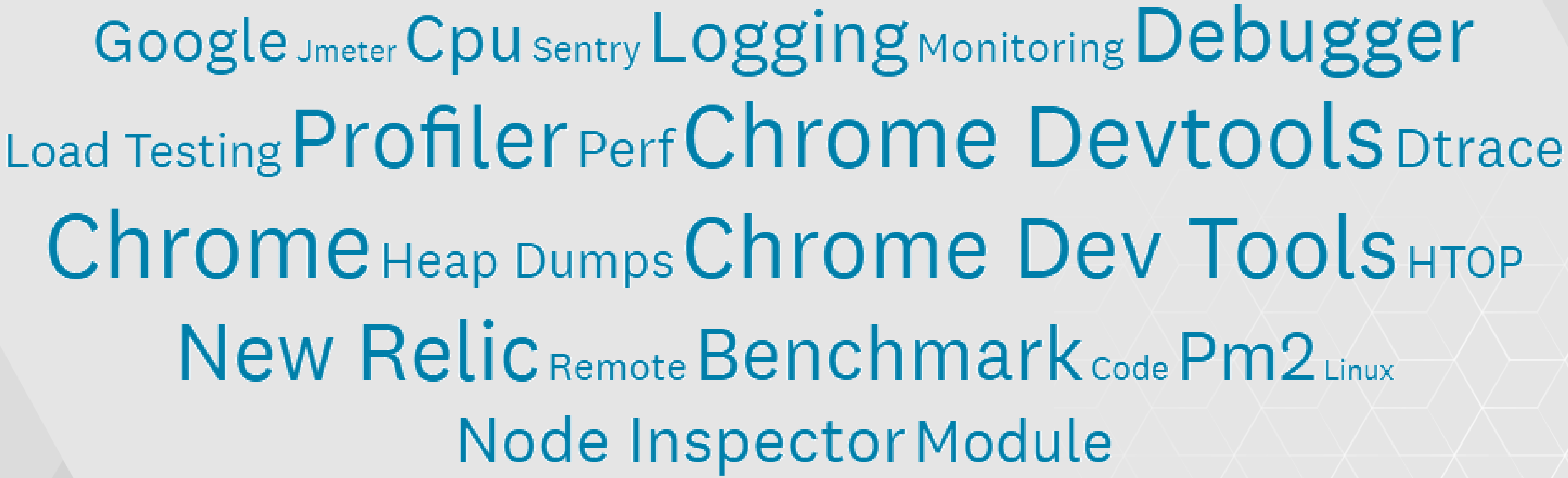
DATA SUMMARY - SUPPLEMENTAL

Q7: If you're tracking performance, which applications do you track performance on?



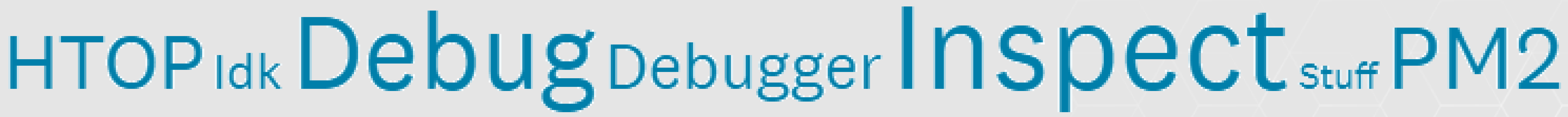
DATA SUMMARY - SUPPLEMENTAL

Q14: What tools do you use to investigate performance issues?

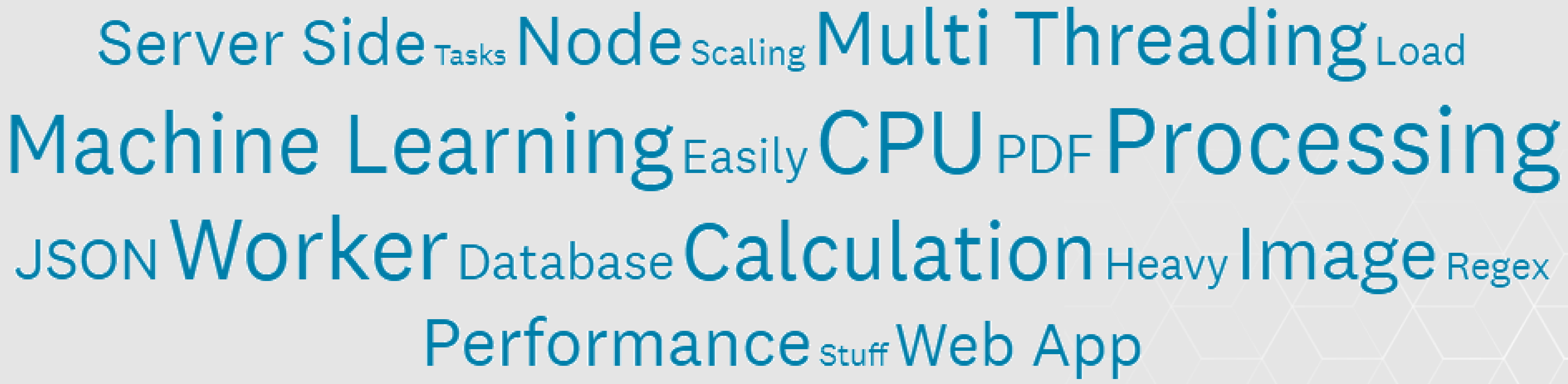


DATA SUMMARY - SUPPLEMENTAL

Q15: What command line options do you use to investigate performance issues?



Q16: Do you have use cases where you cannot use Node.js because it is single threaded? 23% said yes → Q17: Please list the use cases?



Q20: Please list use cases for which startup time is important

Serverless Reboot Container Machine Development Project Load

Electron Startup Testing Deployments Single Starting

Scripts Tools Code Node Dynamic Server Java Services

Cloud CLI Microservice Restarting Webpack API Production

Q24: Which JavaScript language/Node.js features would you like to see optimized next?

Async	7.55%	12
Promises	6.92%	11
Startup	6.29%	10
Modules	5.66%	9
Memory	5.03%	8
Import	3.14%	5
Performance	2.52%	4
Regex	2.52%	4
Faster	2.52%	4
Code	1.89%	3

Thank you

For more information:

Contact ______