Google I/O

Optimizing Dart Applications

+John McCutchan
@johnmccutchan

OBSERVATORY

Photo source info here

View & Evaluate

CPU Profile

Stack trace

Code Coverage

Allocation Profile

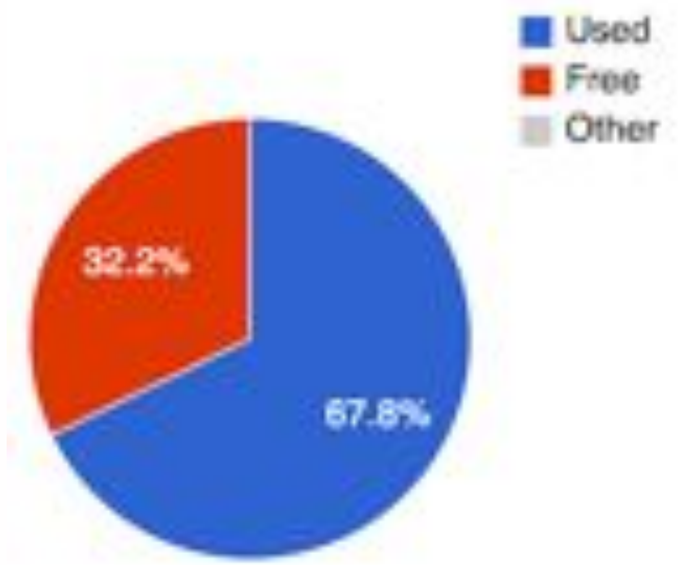
Visual Heap

Code Coverage

```
159         Vector2 a = cc.normal;
160
161         double vRel = a.x * templ.x + a.y * templ.y;
162
163         if (vRel < -Settings.VELOCITY_THRESHOLD) {
164           ccp.velocityBias = -restitution * vRel;
165         }
166       }
167
168       // If we have two points, then prepare the blo
169       if (cc.pointCount == 2) {
170         ContactConstraintPoint ccp1 = cc.points[0];
171         ContactConstraintPoint ccp2 = cc.points[1];
172
173         double invMassA = bodyA.invMass;
174         double invIA = bodyA.invInertia;
175         double invMassB = bodyB.invMass;
176         double invIB = bodyB.invInertia;
177
178         double rn1A = ccp1.rA.cross(cc.normal);
179         double rn1B = ccp1.rB.cross(cc.normal);
180         double rn2A = ccp2.rA.cross(cc.normal);
181         double rn2B = ccp2.rB.cross(cc.normal);
```

Allocation Profile

New Space



Collections	598
Average Collection Time	2.11 ms
Cumulative Collection Time	1.26 secs

Class	Accumulator Size (New) ▼	Accumulator (New)	Current Size (New)	Current (New)	Accumulator Size (Old)	Accumulator (Old)	Current Size (Old)	Current (Old)
_Double	5GB	305,157,620	5MB	349,541	1MB	82,303	1MB	82,057
_Float64Array	3GB	142,156,484	4MB	160,089	453KB	18,417	452KB	18,374
Vector2	684MB	89,590,789	806KB	103,159	133KB	17,045	133KB	17,004
_OneByteString	358MB	17,723,386	421KB	20,691	1MB	21,832	2MB	36,031
_List	282MB	3,832,751	309KB	4,411	2MB	74,243	2MB	44,398
_GrowableList	2MB	119,038	2KB	111	105KB	6,716	24KB	1,504
(dynamic, dynamic) => dynamic	1MB	92,075	1KB	84	0B	0	0B	0
Features	929KB	39,646	0B	0	96KB	4,078	96KB	4,078
ManifoldPoint	929KB	39,640	0B	0	95KB	4,072	95KB	4,072
Matrix2	554KB	70,918	360B	45	11KB	1,370	11KB	1,368
Context	496KB	21,521	600B	25	96B	4	112B	5
_Mint	480KB	30,738	576B	36	48B	3	272B	17
ContactEdge	465KB	19,820	0B	0	48KB	2,036	48KB	2,036
Manifold	465KB	19,820	0B	0	48KB	2,036	48KB	2,036
CircleContact	398KB	8,490	0B	0	40KB	858	40KB	858
ContactID	310KB	39,646	0B	0	32KB	4,078	32KB	4,078
ListIterator	240KB	10,248	264B	11	0B	0	0B	0
SubListIterable	240KB	10,246	264B	11	0B	0	0B	0
Transform	165KB	10,591	192B	12	5KB	346	5KB	344

CPU Profile

VM Tag

User Tag

Exclusive

Inclusive

User Tag

VM Tag

Exclusive

Inclusive

Sampled CPU profile

Timestamp	2014-05-08 09:08:10.928
Time span	2m 11s
Sample count	120000
Sample rate	1000 Hz
Sample depth	8 stack frames
Display cutoff	0.02%
Tags	User > VM ▾

Method	Self
↳ 100.00% Default	100.00%
→ 96.45% Script	96.45%
→ 3.05% GCNewSpace	3.05%
→ 0.38% GCOldSpace	0.38%
→ 0.05% Runtime	0.05%
→ 0.05% Native	0.05%

Tag / function name

Method	Self
↳ 100.00% Default	100.00%
→ 96.45% Script	96.45%
→ 3.05% GCNewSpace	3.05%
→ 0.38% GCOldSpace	0.38%
→ 0.05% Runtime	0.05%
→ 0.05% Native	0.05%

Method	Self
↳ 100.00% Default	100.00%
→ 96.45%	96.45%
→ 3.05%	3.05%
→ 0.38%	0.38%
→ 0.05% Runtime	0.05%
→ 0.05% Native	0.05%

Percentage of parent

Method	Self
↳ 100.00% Default	100.00%
→ 96.45% Script	96.45%
→ 3.05% GCNewSpace	3.05%
→ 0.38% GCOldSpace	0.38%
→ 0.05% Runtime	0.05%
→ 0.05% Native	0.05%

Exclusive

Method	Self
↳ 100.00% Default	100.00%
↳ 96.45% Script	96.45%
→ 8.92% *ContactSolver.solveVelocityConstraints	8.60%
→ 7.44% *ContactSolver.solvePositionConstraints	7.17%
→ 7.34% cos	7.08%

Method	Self
↳ 100.00% Default	100.00%
↳ 96.45% Script	96.45%
→ 8.92% *ContactSolver.solveVelocityConstraints	8.60%
→ 7.44 lvePositionConstraints	7.17%
→ 7.34	7.08%

Percentage of parent

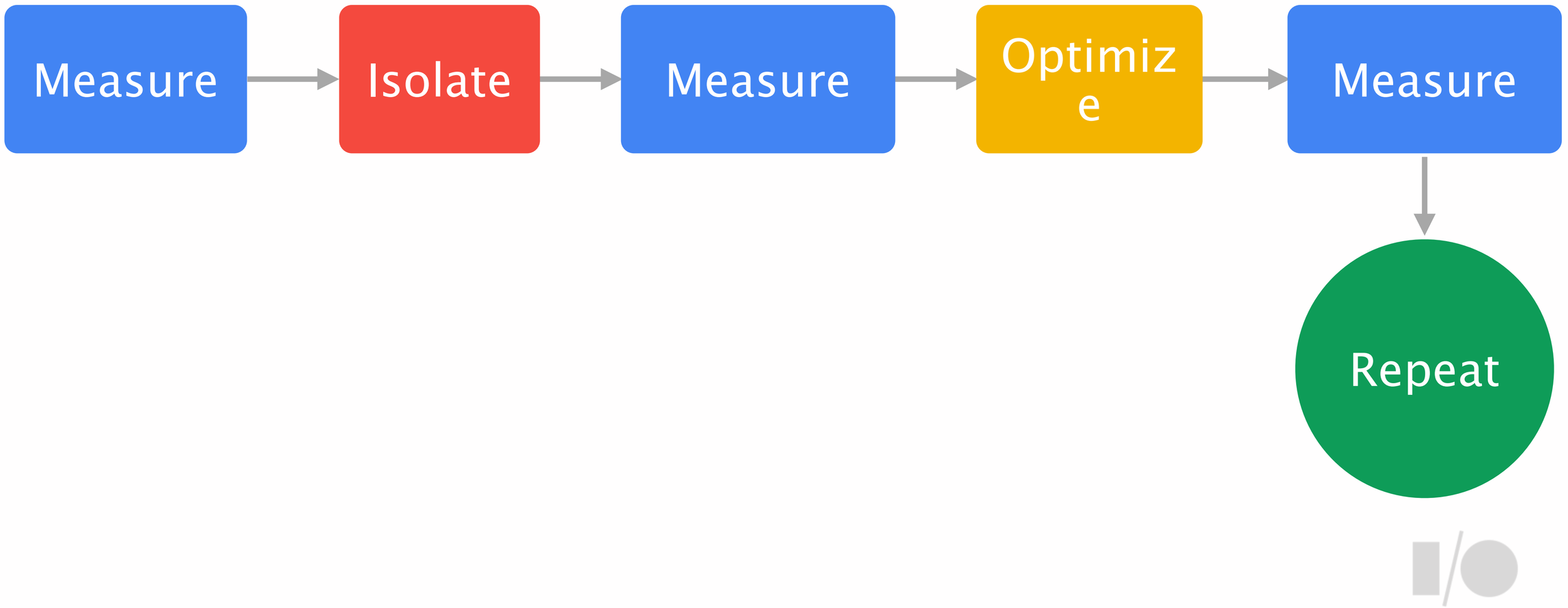
Method	Self
↳ 100.00% Default	100.00%
↳ 96.45% Script	96.45%
→ 8.92% *ContactSolver.solveVelocityConstraints	8.60%
→ 7.44% *ContactSolver.solvePositionConstraints	7.1
→ 7.34% cos	7.0

Exclusive

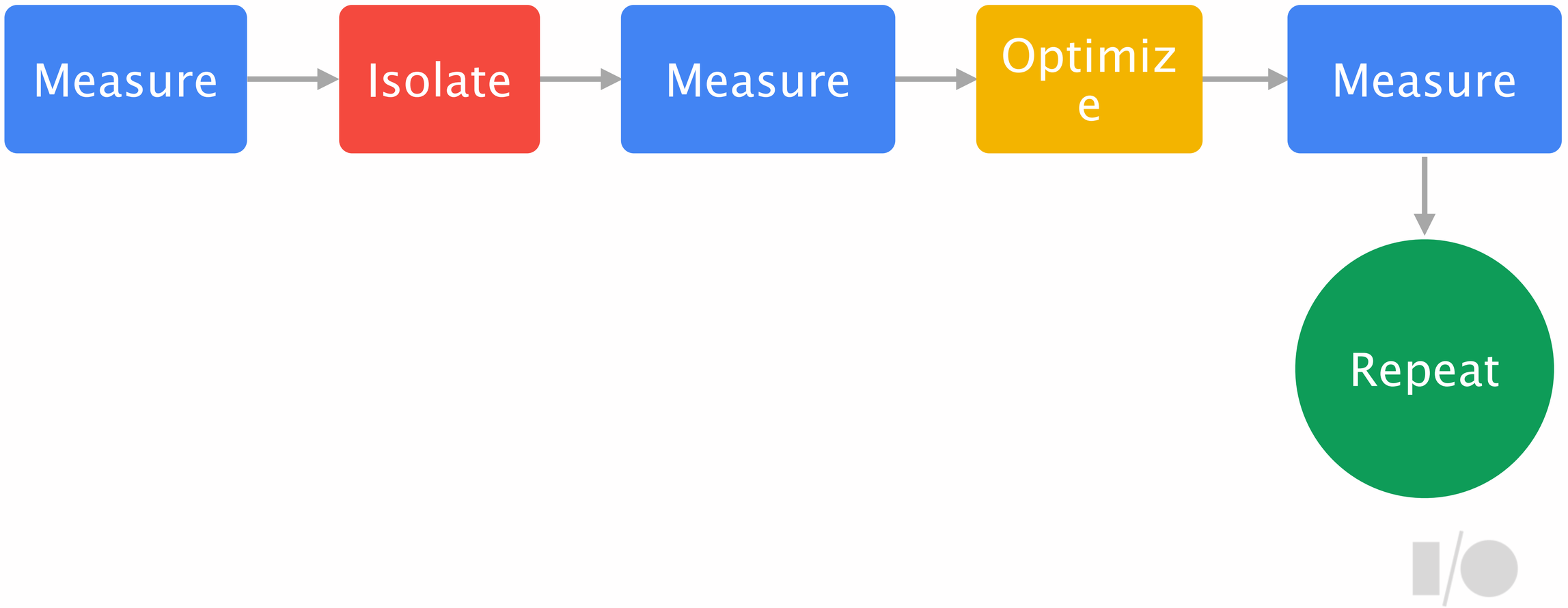
Method	Self
↳ 100.00% Default	100.00%
↳ 96.45% Script	96.45%
→ 8.92% *ContactSolver.solveVelocityConstraints	8.60%
→ 7.44% *ContactSolver.solvePositionConstraints	7.17%
→ 7.34% cos	7.08%

Method	Self
↳ 100.00% Default	100.00%
↳ 96.45% Script	96.45%
↳ 8.92% *ContactSolver.solveVelocityConstraints	8.60%
↳ 100.00% *Island.solve	1.29%
↳ 100.00% *World.solve	1.29%
↳ 100.00% *World.step	0.12%
↳ 54.31% *Box2DBench.step	0.00%
↳ 96.84% Box2DBench.exercise	0.00%
→ 100.00% BenchmarkBase.measure <anonymous closure>	0.00%

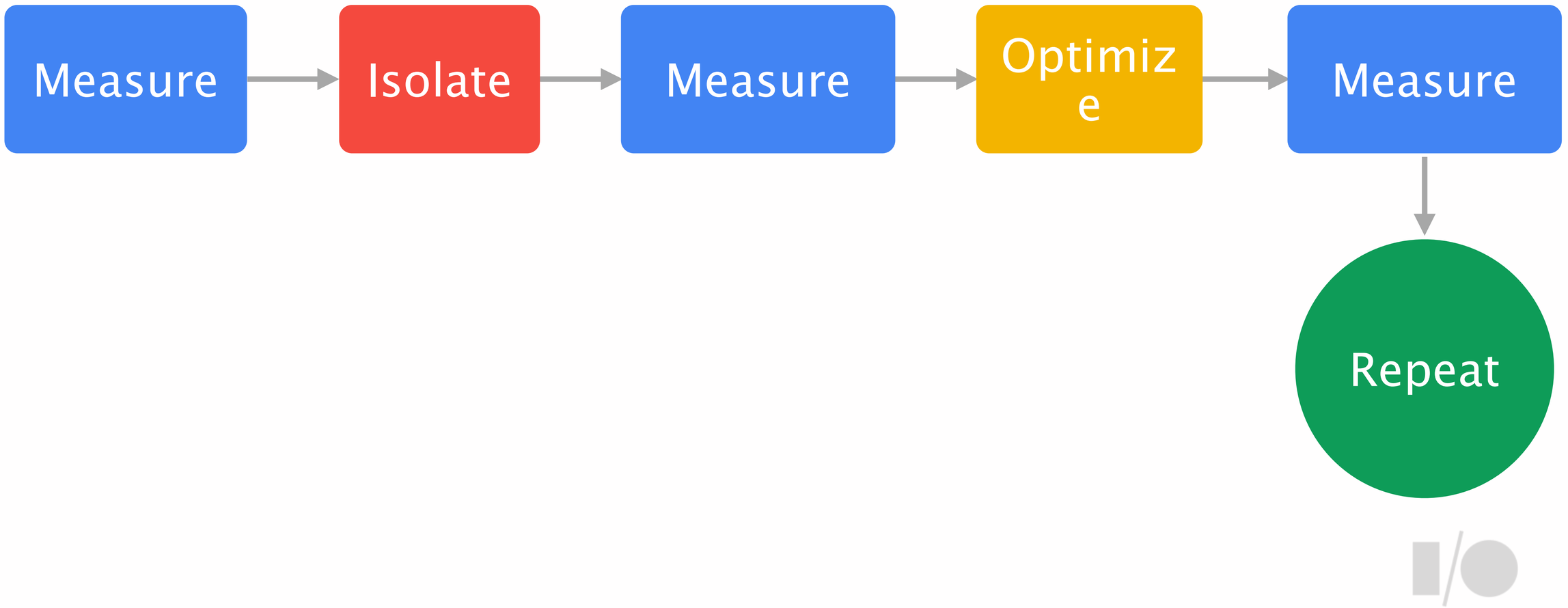
Cycle of Optimization


Measure
Isolate
Measure
Optimiz
e
Measure
Repeat

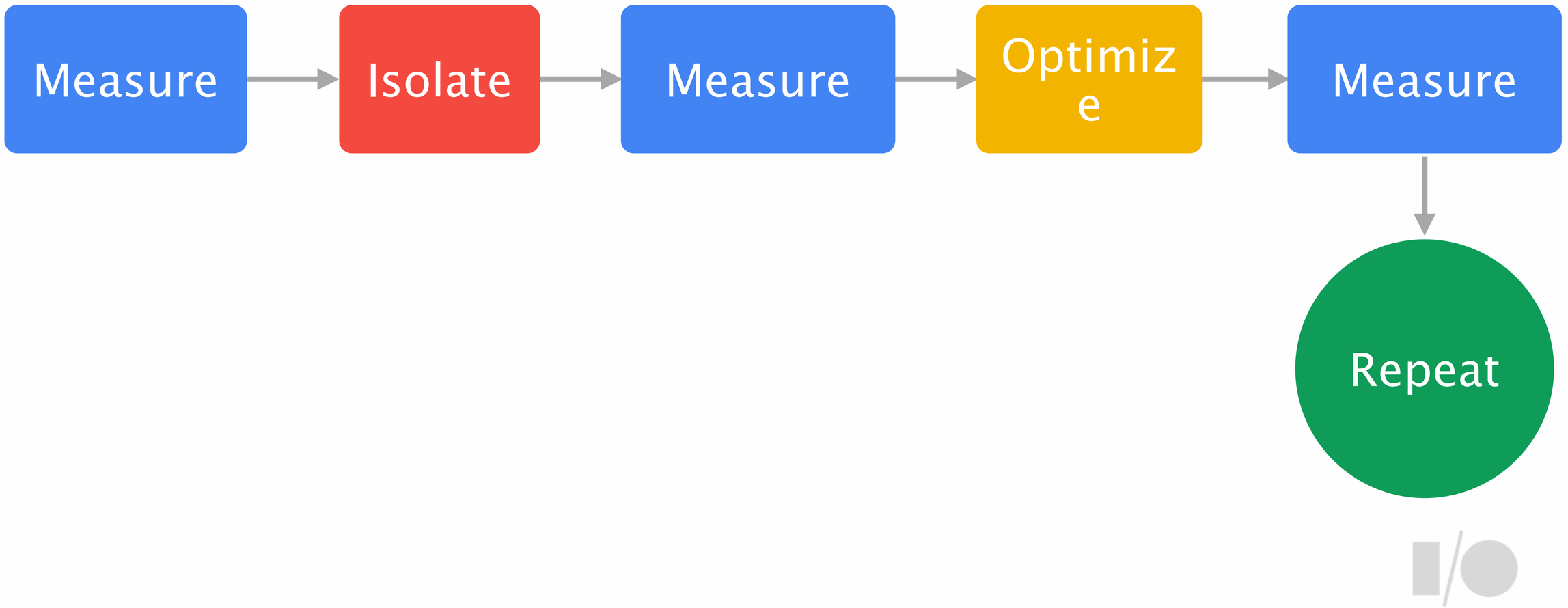
Identify hot spots


Measure
Isolate
Measure
Optimiz
e
Measure
Repeat

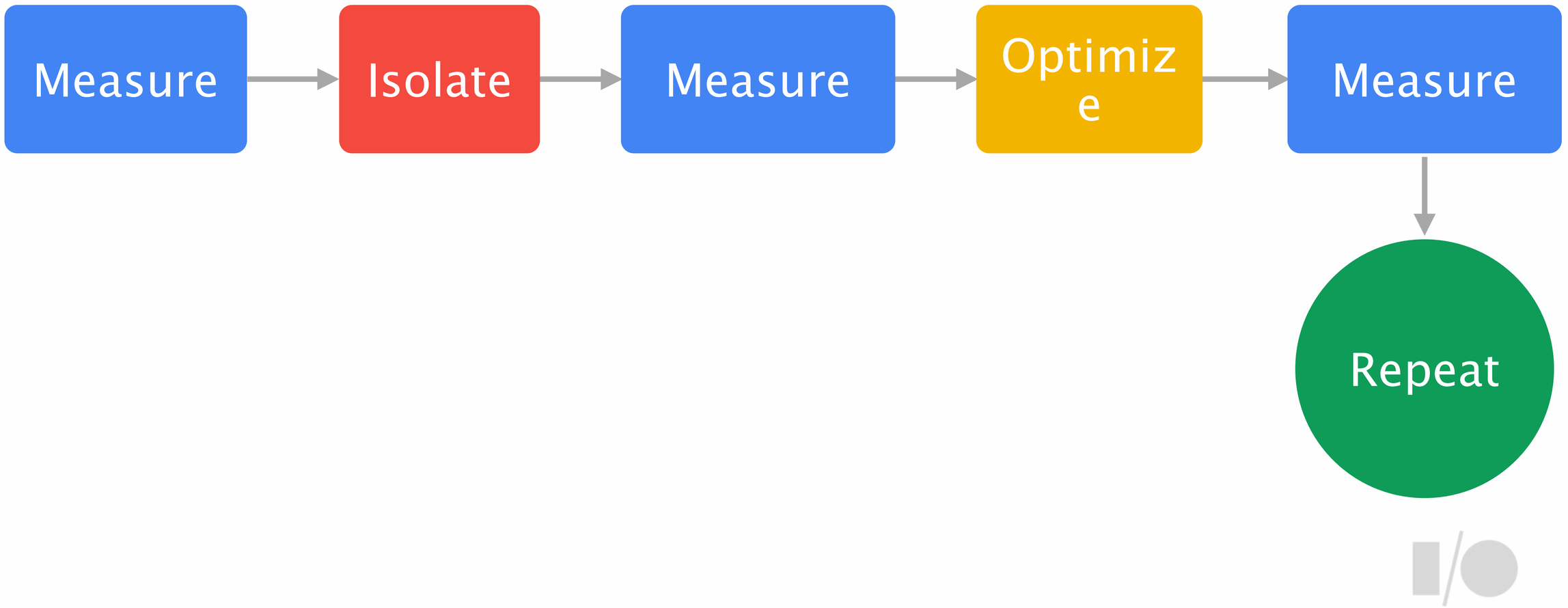
Representative benchmark


Measure
Isolate
Measure
Optimiz
e
Measure
Repeat

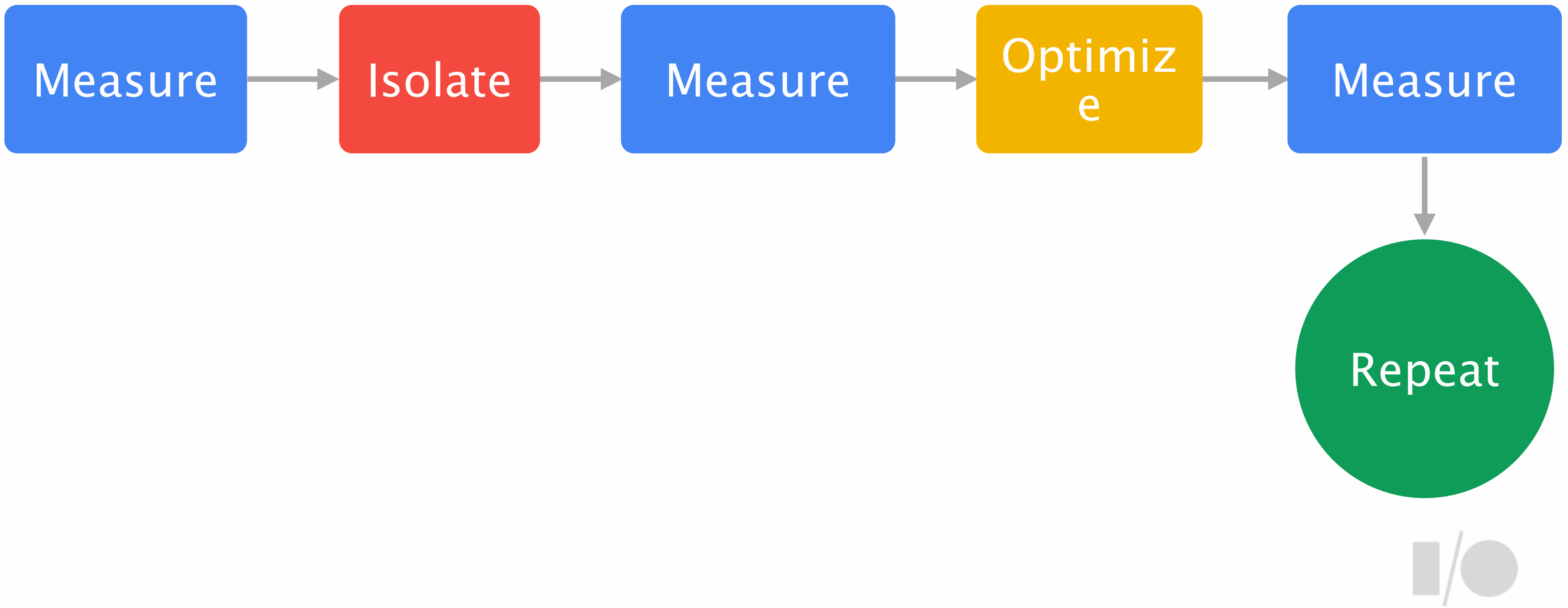
Measure a baseline


Measure
Isolate
Measure
Optimiz
e
Measure
Repeat

Make program run faster….


Measure
Isolate
Measure
Optimiz
e
Measure
Repeat

Compare against baseline


Measure
Isolate
Measure
Optimiz
e
Measure
Repeat

Tips

What to expect and what to look out for

Photo source info here

VM Tags

Script	Executing Dart
VM	Generic VM
GC New Space	Garbage Collecting
GC Old Space	Garbage Collecting
Compile	Compiling Dart
Runtime	C/C++
Native	C/C++

Maximize

Minimiz

Server

User Tag

VM Tag

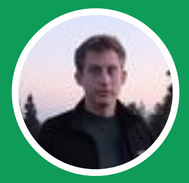
*optimizedFunction

Optimized *

unoptimizedFunc

Thank you!

TODO: optimize
your first Dart app!



+John McCutchan
@johnmccutchan

#perfmatters

www.dartlang.org