

Metrics for the win!

Using Metrics to Understand Application Behavior

Erin Schnabel
@ebullientworks

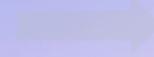
<backstory>

Observability in the Cloud Native era

Orchestrating all the things...

Cloud Environment

- *Dynamic infrastructure*
- Flexible capacity



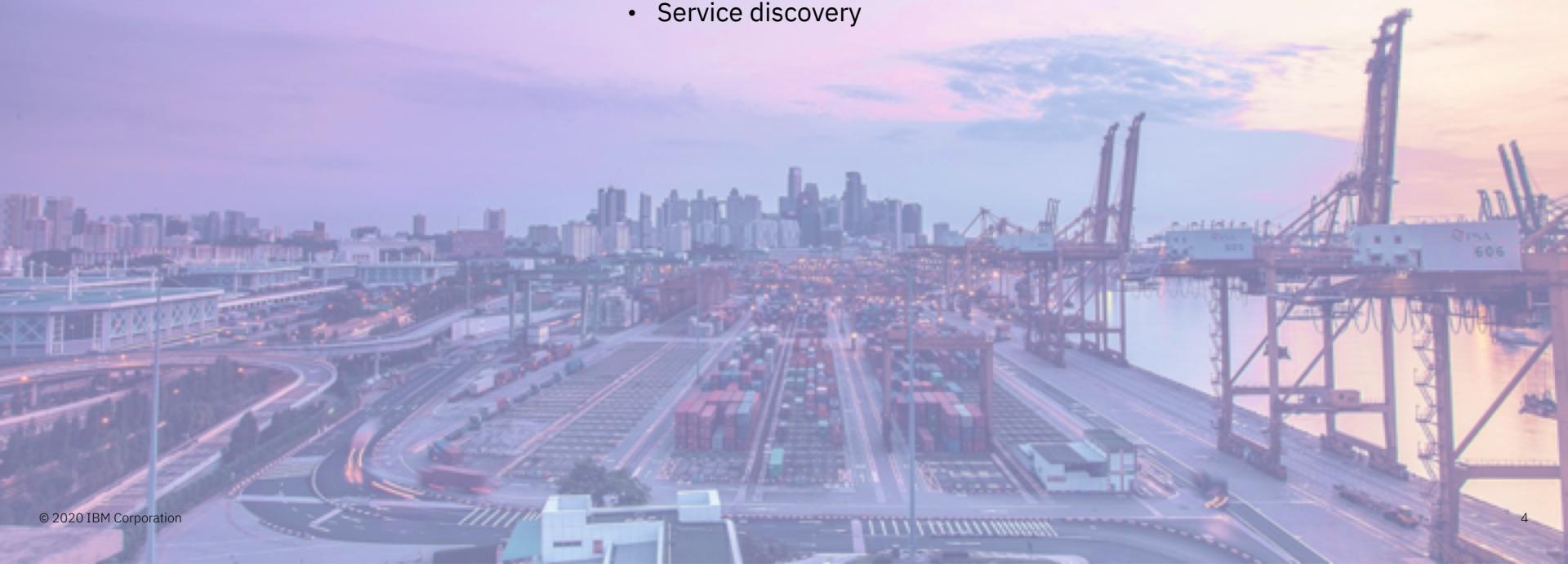
Container orchestration:

- *Self-healing*
- Workload scheduling
- Horizontal scaling
- Service discovery



Containerized workloads

- *Simplified Operations*
- Generic / Transient
- Resource isolation



Orchestration requires Observability

It's 3AM, and things went south...

How much can you understand based
on data you have already collected?

Health Checks



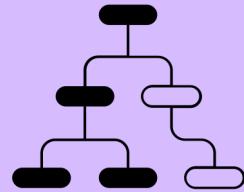
Metrics



Log Entries



Distributed Trace



Which for what?

Service is ready
Service is not a zombie

How many times was
method x called?

What happened when
method x was called

Method x was called

Health Checks



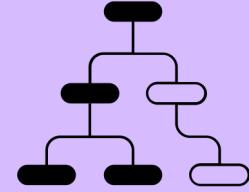
Metrics



Log Entries



Distributed Trace



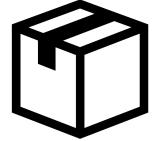
Workload routing
System health

Statistics & trends
Analytics

Service-centric
problem determination

Context + relationships
for end-to-end analysis

Infrastructure



Default probe behaviors

Metrics collection
Resource utilization metrics

Log collection

Span creation & propagation

Health Checks



Specialized probes

Metrics



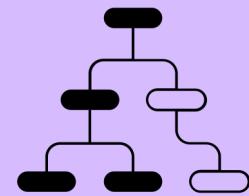
Application-centric
statistical data

Log Entries



Structured or unstructured
text capturing discrete
events

Distributed Trace



Describe relationships
between spans

Application



</backstory>

Metrics #FTW!

Metrics



Prometheus for time series data

Grafana for visualization

App-centric instrumentation

Metrics are not just for ops

MONSTERS!



Added: Ancient Silver Dragon(GARGANTUAN DRAGON){AC:22,HP:487(25d20+225),STR:30(+10),DEX:10(+0),CON:25(+5),INT:12(+2),WIS:12(+1),CHA:12(+1)}
Added: Pteranodon(MEDIUM BEAST){AC:13,HP:13(3dB),STR:12(+1),DEX:15(+2),CON:10(+0),INT:2(-4),WIS:9(-1),CHA:10(+0)}
Added: Adult Bronze Dragon(HUGE DRAGON){AC:19,HP:212(17d12+102),STR:25(+7),DEX:10(+0),CON:23(+6),INT:14(+2),WIS:12(+1),CHA:12(+1)}
Added: Ettercap(MEDIUM MONSTROSITY){AC:13,HP:44(8d8+8),STR:14(+2),DEX:15(+2),CON:13(+1),INT:7(-2),WIS:11(+0),CHA:10(+0)}
Added: Wolf(MEDIUM BEAST){AC:13,HP:11(20d8+2),STR:12(+1),DEX:15(+2),CON:12(+1),INT:3(-4),WIS:12(+1),CHA:10(+0)}
Added: Mummy Lord(MEDIUM UNDEAD){AC:17,HP:97(13d8+39),STR:18(+4),DEX:10(+0),CON:17(+3),INT:11(+0),WIS:12(+1),CHA:12(+1)}
Added: Otyugh(LARGE ABERRATION){AC:14,HP:114(12d10+48),STR:16(+3),DEX:11(+0),CON:19(+4),INT:6(-2),WIS:10(+0),CHA:11(+0)}
Added: Troll(LARGE GIANT){AC:15,HP:84(8d10+40),STR:18(+4),DEX:13(+1),CON:20(+5),INT:7(-2),WIS:9(-1),CHA:12(+1)}
Added: Oni(LARGE GIANT){AC:16,HP:110(13d10+39),STR:19(+4),DEX:11(+0),CON:16(+3),INT:14(+2),WIS:12(+1),CHA:12(+1)}
Added: Vampire Spawn(MEDIUM UNDEAD){AC:15,HP:82(11d8+33),STR:16(+3),DEX:16(+3),CON:16(+3),INT:11(+0),WIS:12(+1),CHA:10(+0)}
Added: Vrock(LARGE FIEND){AC:15,HP:104(11d10+44),STR:17(+3),DEX:15(+2),CON:18(+4),INT:8(-1),WIS:13(+0),CHA:12(+1)}
Added: Werebear(MEDIUM HUMANOID){AC:11,HP:135(18d8+54),STR:19(+4),DEX:10(+0),CON:17(+3),INT:11(+0),WIS:12(+1),CHA:11(+0)}
Added: Quasit(TINY FIEND){AC:13,HP:7(3d4),STR:5(-3),DEX:17(+3),CON:10(+0),INT:7(-2),WIS:10(+0),CHA:10(+0)}
Added: Wraith(MEDIUM UNDEAD){AC:13,HP:67(9d8+27),STR:6(-2),DEX:16(+3),CON:16(+3),INT:12(+1),WIS:14(+2),CHA:11(+0)}
Added: Young Brass Dragon(LARGE DRAGON){AC:17,HP:110(13d10+39),STR:19(+4),DEX:10(+0),CON:17(+3),INT:11(+0),WIS:12(+1),CHA:12(+1)}
Added: Iron Golem(LARGE CONSTRUCT){AC:20,HP:210(20d10+100),STR:24(+7),DEX:9(-1),CON:20(+5),INT:3(-4),WIS:12(+1),CHA:12(+1)}
Added: Adult Blue Dragon(HUGE DRAGON){AC:19,HP:225(18d12+108),STR:25(+7),DEX:10(+0),CON:23(+6),INT:10(+0),WIS:13(+0),CHA:13(+0)}
Added: Pit Fiend(LARGE FIEND){AC:19,HP:300(24d10+168),STR:26(+8),DEX:14(+2),CON:24(+7),INT:22(+6),WIS:14(+0),CHA:14(+0)}
Added: Unicorn(LARGE CELESTIAL){AC:12,HP:67(9d10+18),STR:18(+4),DEX:14(+2),CON:15(+2),INT:11(+0),WIS:13(+0),CHA:13(+0)}
Added: Black Dragon Wyrmling(MEDIUM DRAGON){AC:17,HP:33(6d8+6),STR:15(+2),DEX:14(+2),CON:13(+1),INT:10(+0),WIS:12(+1),CHA:12(+1)}
Added: Adult White Dragon(HUGE DRAGON){AC:18,HP:200(16d12+96),STR:22(+6),DEX:10(+0),CON:22(+6),INT:8(+0),WIS:13(+0),CHA:13(+0)}
Added: Ancient Copper Dragon(GARGANTUAN DRAGON){AC:21,HP:350(20d20+140),STR:27(+8),DEX:12(+1),CON:25(+5),INT:12(+2),WIS:14(+0),CHA:14(+0)}
Added: Treant(HUGE PLANT){AC:16,HP:138(12d12+60),STR:23(+6),DEX:8(-1),CON:21(+5),INT:12(+1),WIS:16(+4),CHA:16(+4)}
Added: Azer(MEDIUM ELEMENTAL){AC:17,HP:39(6d8+12),STR:17(+3),DEX:12(+1),CON:15(+2),INT:12(+1),WIS:13(+0),CHA:13(+0)}
Added: Young White Dragon(LARGE DRAGON){AC:17,HP:133(14d10+56),STR:18(+4),DEX:10(+0),CON:18(+4),INT:10(+0),WIS:12(+1),CHA:12(+1)}
Added: Pegasus(LARGE CELESTIAL){AC:12,HP:59(7d10+21),STR:18(+4),DEX:15(+2),CON:16(+3),INT:10(+0),WIS:13(+0),CHA:13(+0)}
Added: Ancient Brass Dragon(GARGANTUAN DRAGON){AC:20,HP:297(17d20+119),STR:27(+8),DEX:10(+0),CON:25(+5),INT:13(+2),WIS:14(+0),CHA:14(+0)}
Added: Lich(MEDIUM UNDEAD){AC:17,HP:135(18d8+54),STR:11(+0),DEX:16(+3),CON:16(+3),INT:20(+5),WIS:14(-2),CHA:14(-2)}
Added: Skeleton(MEDIUM UNDEAD){AC:13,HP:13(2d8+4),STR:10(+0),DEX:14(+2),CON:15(+2),INT:6(-2),WIS:8(-1),CHA:8(-1)}
Added: Adult Red Dragon(HUGE DRAGON){AC:19,HP:256(19d12+133),STR:27(+8),DEX:10(+0),CON:25(+7),INT:16(+0),WIS:15(+0),CHA:15(+0)}
Added: Tyrannosaurus Rex(HUGE BEAST){AC:13,HP:136(13d12+52),STR:25(+7),DEX:10(+0),CON:19(+4),INT:2(-4),WIS:12(+1),CHA:12(+1)}
Added: Aarakocra(MEDIUM HUMANOID){AC:12,HP:13(3d8),STR:10(+0),DEX:14(+2),CON:10(+0),INT:11(+0),WIS:12(+1),CHA:12(+1)}
Added: Triceratops(HUGE BEAST){AC:13,HP:95(10d12+30),STR:22(+6),DEX:9(-1),CON:17(+3),INT:2(-4),WIS:11(+0),CHA:11(+0)}
Added: Gelatinous Cube(LARGE OOZE){AC:6,HP:84(8d10+40),STR:14(+2),DEX:3(-4),CON:20(+5),INT:1(-5),WIS:10(+0),CHA:10(+0)}
Added: Hell Hound(MEDIUM FIEND){AC:15,HP:45(7d8+14),STR:17(+3),DEX:12(+1),CON:14(+2),INT:6(-2),WIS:11(+0),CHA:11(+0)}
Added: Planetary(LARGE CELESTIAL){AC:19,HP:200(16d10+112),STR:24(+7),DEX:20(+5),CON:24(+7),INT:19(-4),WIS:13(+0),CHA:13(+0)}
Added: Homunculus(TINY CONSTRUCT){AC:13,HP:5(5d24),STR:4(-3),DEX:15(+2),CON:11(+0),INT:10(+0),WIS:10(-1),CHA:10(-1)}
Added: Ancient Black Dragon(GARGANTUAN DRAGON){AC:22,HP:367(21d20+147),STR:27(+8),DEX:14(+2),CON:25(+5),INT:14(+2),WIS:15(+0),CHA:15(+0)}
Added: Adult Brass Dragon(HUGE DRAGON){AC:18,HP:172(15d12+75),STR:23(+6),DEX:10(+0),CON:21(+5),INT:14(+0),WIS:13(+0),CHA:13(+0)}
Added: Ancient Bronze Dragon(GARGANTUAN DRAGON){AC:22,HP:444(24d20+192),STR:29(+9),DEX:10(+0),CON:27(+6),INT:17(+5),WIS:16(+0),CHA:16(+0)}
Added: Glabrezu(LARGE FIEND){AC:17,HP:157(15d10+75),STR:20(+5),DEX:15(+2),CON:21(+5),INT:19(+4),WIS:14(+0),CHA:14(+0)}
Added: Lamia(LARGE MONSTROSITY){AC:13,HP:97(13d10+26),STR:16(+3),DEX:13(+1),CON:15(+2),INT:14(+2),WIS:12(+1),CHA:12(+1)}
Added: Bulette(LARGE MONSTROSITY){AC:17,HP:94(9d10+45),STR:19(+4),DEX:11(+0),CON:21(+5),INT:2(-4),WIS:11(+0),CHA:11(+0)}
Added: Young Black Dragon(LARGE DRAGON){AC:18,HP:127(15d10+45),STR:19(+4),DEX:14(+2),CON:17(+3),INT:10(+0),WIS:13(+0),CHA:13(+0)}
Added: Dryad(MEDIUM FEY){AC:11,HP:22(5d8),STR:10(+0),DEX:12(+1),CON:11(+0),INT:14(+2),WIS:15(+2),CHA:15(+2)}
Added: Elephant(HUGE BEAST){AC:12,HP:76(8d12+24),STR:22(+6),DEX:9(-1),CON:17(+3),INT:3(-4),WIS:11(+0),CHA:12(+0)}
Added: Deva(MEDIUM CELESTIAL){AC:17,HP:136(16d8+64),STR:18(+4),DEX:18(+4),CON:18(+4),INT:17(+39),WIS:16(+0),CHA:16(+0)}
Added: Weretiger(MEDIUM HUMANOID){AC:12,HP:120(16d8+48),STR:17(+3),DEX:15(+2),CON:16(+3),INT:10(+0),WIS:12(+1),CHA:12(+1)}

Monsters & Highlander

(there can be only one)

Dungeons & Dragons inspired encounters

From scraped wiki pages → ~**215 usable monsters**

Monster has **Type**, **Size**, **Armor Class**, **Hit Points**:

Dryad(MEDIUM FEY)	AC:11, HP:22(5d8) d12+24
Deva(MEDIUM CELESTIAL)	AC:17, HP:136(16d8+64)
Weretiger(MEDIUM HUMANOID)	AC:12 , HP:120(16d8+48)
Pseudodragon(TINY DRAGON)	AC:13, HP:7(2d4+2)

Monsters use weapons

Each has a **type**, and inflicts variable **damage**

Ancient Green Dragon:

bite:	piercing ,	19(2d10+8)
claw:	slashing,	19(2d10+8)
tail:	bludgeoning,	17(2d8+8)
poison breath:	poison,	77(22d6)

→ multiattack (e.g. 2 claws and 1 bite) preferred

Monsters & Highlander

(there can be only one)

Dungeons & Dragons inspired encounters

SO. MANY. INTERESTING. QUESTIONS.

1. Monsters engage in **encounters**:

- /combat/**any** 2-5 opponents

2. Combat encounter has **stages**:

1. Check for **surprise**: Dexterity vs. Perception *
2. Roll for **initiative**
3. Repeat **rounds** (in initiative order) until only one left

Some points of view:

- A) Volunteer DM just wants to know how stuff works
- B) Game designer wants to understand how changes in algorithms impact game play.
- C) Developer wants to know app works as expected.

Time-series data

String key with *ONE** numeric value

Value is *observed* at collection time

Periodic collection over time → series
Each value *appended* as new entry

Time is a primary axis (x)



Useful for spotting:

- Trends
- Variability
- Rate of change

* “Univariate”. Keeps things from getting out of hand.

Dimensional data

Consider **hierarchical** data (our old friend):

```
app.memory.used
```

Add datacenter, and instance id:

```
datacenter.instance_id.app.memory.used  
*.instance_id.app.memory.used  
**.app.memory.used
```

What if you want to add

- environment (dev, test, prod)
- service name

Tags / Labels add **dimensions** for analysis:

```
app.memory.used {  
    datacenter=us1  
    instance_id=pod1234  
    env=prod  
    service=serviceName  
}
```

Filtered aggregation of single numeric value

Not constrained by wildcards / hierarchy

Cardinality explosion

New series for each unique combination

Label A (10 values), Label B (100 values),

Add pod (p), instance (i), service (s)

(10 * 100 * p * i * s) unique time series!

→ **Normalize and bound dimensions**

- attackName="Bite" (bite or bites, pick one)
- attackType="piercing"

Real (bad) examples of unbounded values:

- 404 URLs
- request ids
- user ids

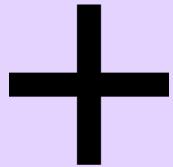
NUMBERS!

```
encounter_rounds_max{crDelta="17",numCombatants="05",sizeDelta="03",targetSelector="Random",} 0.0
encounter_rounds_max{crDelta="20",numCombatants="06",sizeDelta="03",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="08",numCombatants="05",sizeDelta="01",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="01",numCombatants="02",sizeDelta="01",targetSelector="FaceOff",} 0.0
encounter_rounds_max{crDelta="05",numCombatants="05",sizeDelta="02",targetSelector="LowestHealth",} 0.0
encounter_rounds_max{crDelta="10",numCombatants="03",sizeDelta="00",targetSelector="HighestHealth",} 0.0
encounter_rounds_max{crDelta="11",numCombatants="06",sizeDelta="03",targetSelector="HighestHealth",} 0.0
encounter_rounds_max{crDelta="19",numCombatants="06",sizeDelta="04",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="08",numCombatants="05",sizeDelta="04",targetSelector="BiggestFirst",} 0.0
encounter_rounds_max{crDelta="21",numCombatants="05",sizeDelta="02",targetSelector="BiggestFirst",} 0.0
encounter_rounds_max{crDelta="18",numCombatants="03",sizeDelta="01",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="09",numCombatants="03",sizeDelta="02",targetSelector="HighestHealth",} 0.0
encounter_rounds_max{crDelta="21",numCombatants="04",sizeDelta="03",targetSelector="HighestHealth",} 0.9
encounter_rounds_max{crDelta="16",numCombatants="06",sizeDelta="03",targetSelector="LowestHealth",} 0.0
encounter_rounds_max{crDelta="15",numCombatants="03",sizeDelta="00",targetSelector="LowestHealth",} 0.0
encounter_rounds_max{crDelta="08",numCombatants="05",sizeDelta="02",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="20",numCombatants="06",sizeDelta="04",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="01",numCombatants="02",sizeDelta="00",targetSelector="FaceOff",} 0.0
encounter_rounds_max{crDelta="17",numCombatants="05",sizeDelta="02",targetSelector="Random",} 0.0
encounter_rounds_max{crDelta="27",numCombatants="03",sizeDelta="03",targetSelector="Random",} 0.0
encounter_rounds_max{crDelta="18",numCombatants="03",sizeDelta="04",targetSelector="BiggestFirst",} 0.0
encounter_rounds_max{crDelta="26",numCombatants="04",sizeDelta="03",targetSelector="LowestHealth",} 0.0
encounter_rounds_max{crDelta="11",numCombatants="06",sizeDelta="04",targetSelector="HighestHealth",} 0.0
encounter_rounds_max{crDelta="05",numCombatants="05",sizeDelta="01",targetSelector="LowestHealth",} 0.0
encounter_rounds_max{crDelta="10",numCombatants="03",sizeDelta="01",targetSelector="HighestHealth",} 0.0
encounter_rounds_max{crDelta="06",numCombatants="04",sizeDelta="00",targetSelector="Random",} 0.0
encounter_rounds_max{crDelta="09",numCombatants="04",sizeDelta="00",targetSelector="BiggestFirst",} 0.0
encounter_rounds_max{crDelta="30",numCombatants="04",sizeDelta="03",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="21",numCombatants="05",sizeDelta="01",targetSelector="BiggestFirst",} 0.0
encounter_rounds_max{crDelta="09",numCombatants="03",sizeDelta="03",targetSelector="HighestHealth",} 0.0
encounter_rounds_max{crDelta="18",numCombatants="03",sizeDelta="02",targetSelector="SmallestFirst",} 0.0
encounter_rounds_max{crDelta="19",numCombatants="06",sizeDelta="05",targetSelector="SmallestFirst",} 0.0
dice_rolls_max{die="d10",face="08",} 8.0
dice_rolls_max{die="d10",face="09",} 9.0
dice_rolls_max{die="d10",face="06",} 6.0
dice_rolls_max{die="d10",face="07",} 7.0
dice_rolls_max{die="d10",face="04",} 4.0
dice_rolls_max{die="d10",face="05",} 5.0
dice_rolls_max{die="d10",face="02",} 2.0
round_attacks_max{attackName="Bite",attackType="piercing",attacker="Sahuagin",hitOrMiss="hit",targetSe
round_attacks_max{attackName="Bite",attackType="piercing",attacker="Red Dragon Wyrmling",hitOrMiss="hi
round_attacks_max{attackName="Crush",attackType="bludgeoning",attacker="Darkmantle",hitOrMiss="hit",ta
round_attacks_max{attackName="Bite",attackType="piercing",attacker="Adult Silver Dragon",hitOrMiss="mi
round_attacks_max{attackName="Pseudopod",attackType="acid",attacker="Gelatinous Cube",hitOrMiss="critic
round_attacks_max{attackName="Greatsword",attackType="slashing",attacker="Planetary",hitOrMiss="hit",ta
round_attacks_max{attackName="Claw",attackType="slashing",attacker="Ancient White Dragon",hitOrMiss="mi
round_attacks_max{attackName="Beak",attackType="piercing",attacker="Owlbear",hitOrMiss="critical",target
round_attacks_max{attackName="Pike",attackType="piercing",attacker="Centaur",hitOrMiss="miss",targetSel
round_attacks_max{attackName="Longsword",attackType="slashing",attacker="Hobgoblin",hitOrMiss="critica
```

Meter types: Micrometer

Incrementing value

Counter



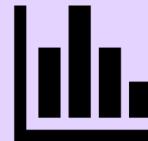
Observed value

Gauge



Distribution of values (buckets)

Histogram



Bucket x Timer

Summary



Counter

Gauge

DistributionSummary

DistributionSummary

Timer

count, sum, max

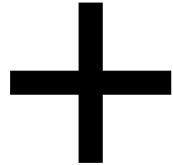
Duration (total time)

AND

Distribution summary

Optional: percentiles, SLAs, buckets

Rolling the dice



Static class for Dice, as hub for all things random.

“Monitor” injection point (to allow for different metrics systems)

With micrometer, e.g., rolls are **counted** like this:

```
Dice.setMonitor((k, v) -> registry.counter("dice.rolls", "die", k, "face", label(v)).increment());
```

```
# HELP dice_rolls_total
# TYPE dice_rolls_total counter
dice_rolls_total{die="d10",face="08"} 16750.0
dice_rolls_total{die="d10",face="09"} 16724.0
dice_rolls_total{die="d10",face="06"} 16804.0
dice_rolls_total{die="d10",face="07"} 16883.0
dice_rolls_total{die="d10",face="04"} 16944.0
dice_rolls_total{die="d10",face="05"} 16695.0
dice_rolls_total{die="d10",face="02"} 16960.0
dice_rolls_total{die="d10",face="03"} 16843.0
dice_rolls_total{die="d10",face="10"} 17034.0
dice_rolls_total{die="d20",face="18"} 24008.0
dice_rolls_total{die="d20",face="19"} 24120.0
dice_rolls_total{die="d20",face="16"} 24106.0
dice_rolls_total{die="d20",face="17"} 23922.0
```

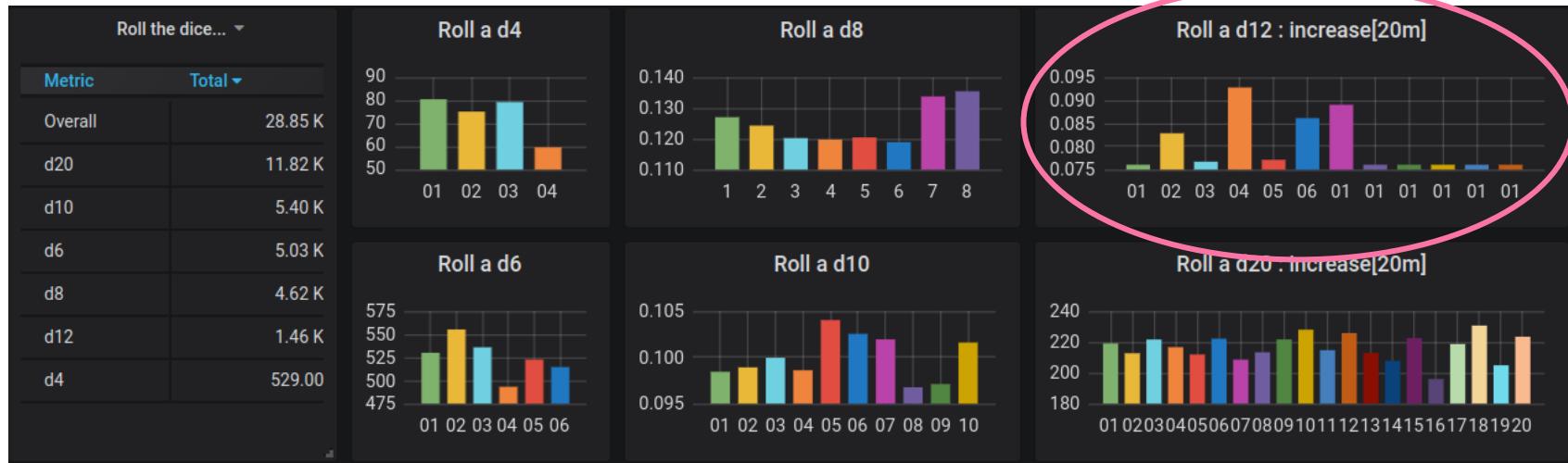
How often was a 9 rolled on a d10: —

,instance="quarkus:8080",job="quarkus_micrometer"

Roll the dice . . .

All rolls of the dice: total count, or rate/increase

(“early” version, so they aren’t all consistent, the labels are even messed up!)



What is wrong with this picture?

Roll the dice ...

All rolls of the dice: total count, or rate/increase

“...the more I learn, the more I realize how little I know.” — John D. Rockefeller

Labels are even messed up!)

Metric
Overall
d20
d10
d6
d8
d12
d4

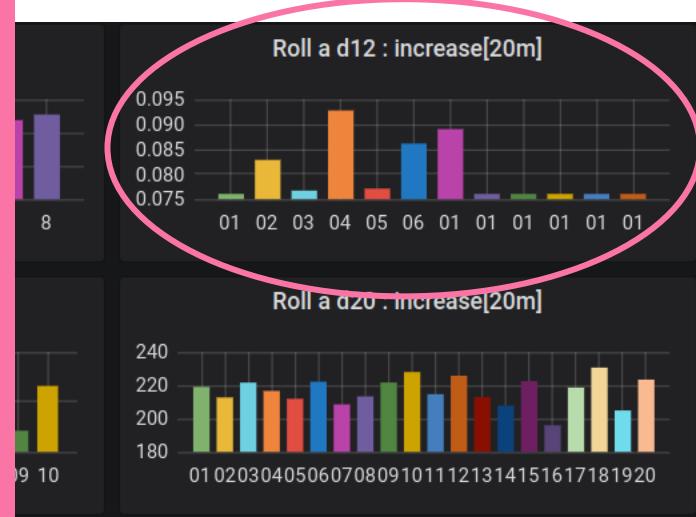
Straight-up bug not caught in unit test.

(16d12+80)

vs

17(2d10+6)

I tested that damage phrases could be parsed,
but I did not have a test for each die.



ture?

Roll the dice . . . Correctly, this time

Take care with scale!

▼ Roll the dice

Roll the dice...

Metric	Total ▾
Overall	780.24 K
d20	364.64 K
d10	127.66 K
d6	123.04 K
d8	113.57 K
d12	42.52 K
d4	8.82 K

Roll a d20 : rate



Roll a d10 : rate



Roll a d6 : rate



Roll a d12 : rate



Roll a d8 : rate



Roll a d4 : rate



sum by (face) (rate(dice_rolls_total{die="d12"}[\$__interval]))

Each graph focuses on frequency of rolls for each die within a fixed interval.
The y-axis gives an indication of comparative frequency.

Attacks : Hits and Misses

How many attacks are successful?

- **20 – critical hit (HIT)**
- **1 – critical miss (MISS)**

• Otherwise, EITHER

- Attack modifier vs. Armor Class

OR

- Saving Throw against Spell DC

: oneRound:

Troll(LARGE GIANT){**AC:15**,HP:84(8d10+40),STR:18(+4),DEX:13(+1),CON:20(+5),INT:7(-2),WIS:

Pit Fiend(LARGE FIEND){**AC:19**,HP:300(24d10+168),STR:26(+8),DEX:14(+2),CON:24(+7),INT:22(+6),WIS:19(+4),CHA:10(+2)}

: attack: **miss**: Troll(36) -> Pit Fiend(100)

: attack: **miss**: Troll(36) -> Pit Fiend(100)

: attack: **hit**> Troll(36) -> Pit Fiend(97) for 9 damage using **Claws[7hit,11(2d6+4)|slashing]**

: attack: **hit**> Pit Fiend(97) -> Troll(10) for 22 damage using **Bite[14hit,22(4d6+8)|piercing]**

: attack: **MISS**: Pit Fiend(97) -> Troll(10)

: attack: **HIT**> Pit Fiend(97) -> Troll(0) for 34 damage using **Mace[14hit,15(2d6+8)|bludgeoning]**

: oneRound: survivors

Pit Fiend(LARGE FIEND){AC:19,HP:300(24d10+168),STR:26(+8),DEX:14(+2),CON:24(+7),INT:22(+6),WIS:19(+4),CHA:10(+2)}

Attacks: Hits and Misses



```
registry.summary("attack.success",
    "hitOrMiss", event.hitOrMiss())
    .record((double) event.getDifficultyClass() - event.getAttackModifier());
```

```
# HELP attack_success
# TYPE attack_success summary
attack_success_count{hitOrMiss="saved hit",} 2537.0
attack_success_sum{hitOrMiss="saved hit",} 21397.0
attack_success_count{hitOrMiss="hit",} 87472.0
attack_success_sum{hitOrMiss="hit",} 625735.0
attack_success_count{hitOrMiss="critical miss",} 7524.0
attack_success_sum{hitOrMiss="critical miss",} 63263.0
attack_success_count{hitOrMiss="miss",} 48710.0
attack_success_sum{hitOrMiss="miss",} 529061.0
attack_success_count{hitOrMiss="critical hit",} 7488.0
attack_success_sum{hitOrMiss="critical hit",} 63263.0
```

```
# HELP attack_success_max
# TYPE attack_success_max gauge
attack_success_max{hitOrMiss="saved hit",} 6.0
attack_success_max{hitOrMiss="hit",} 18.0
attack_success_max{hitOrMiss="critical miss",} 15.0
attack_success_max{hitOrMiss="miss",} 20.0
attack_success_max{hitOrMiss="critical hit",} 15.0
```

How many “saved hits” (count)

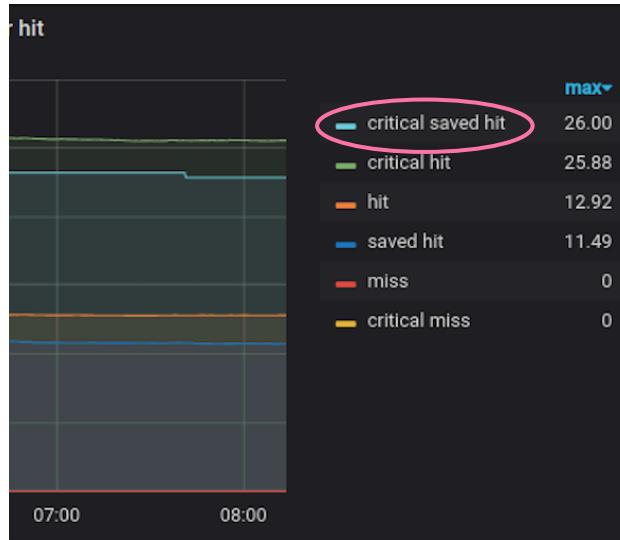
Accumulated value (sum):

event.getDifficultyClass()
–
event.getAttackModifier()

Maximum individual value:

event.getDifficultyClass()
–
event.getAttackModifier()

Attacks: Hits and Misses



Another early graph of an early/partial run

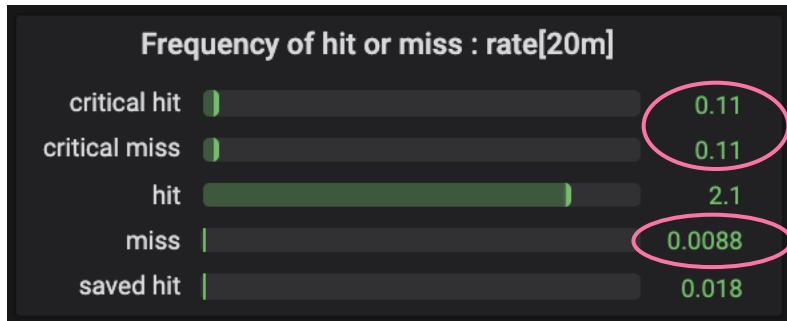
There are a few things wrong, but let's start with..

A *critical saved hit* is not a thing.

Value is a combination of Booleans into one string

```
public String hitOrMiss() {  
    return (critical ? "critical " : "")  
        + (saved ? "saved " : "")  
        + (hit ? "hit" : "miss");  
}
```

Attacks: Hits and Misses



This view didn't last long, but it highlighted another problem.

Can you spot it?

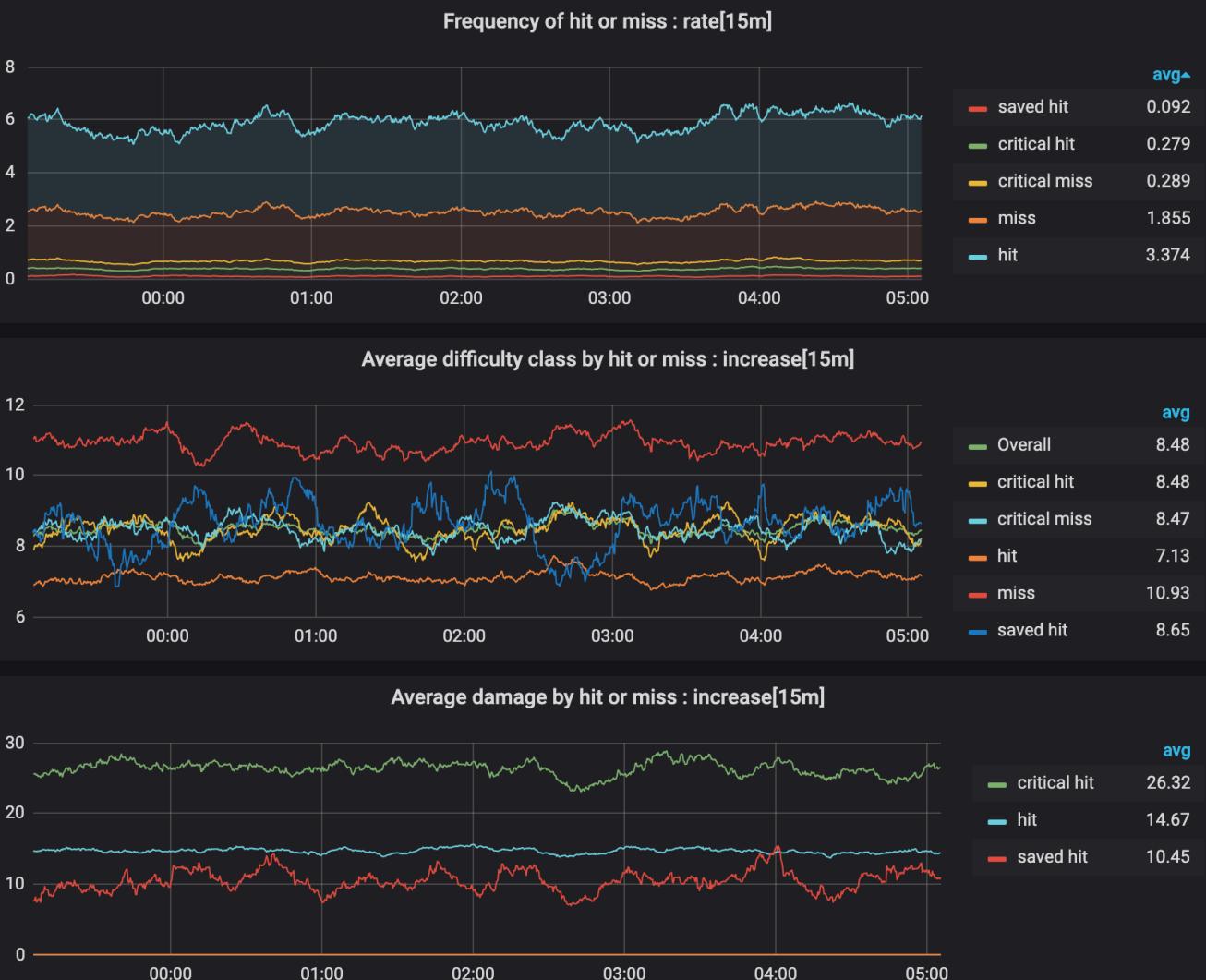
Misses should be _way_ more common.

This ended up being a problem with the original conversion from HTML to JSON.

Attacks: Hits and Misses

Correctly,
this time

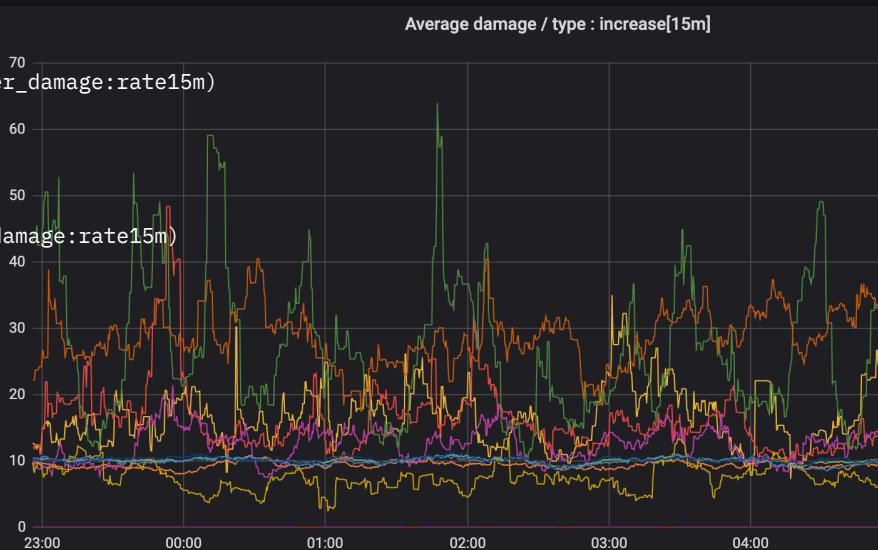
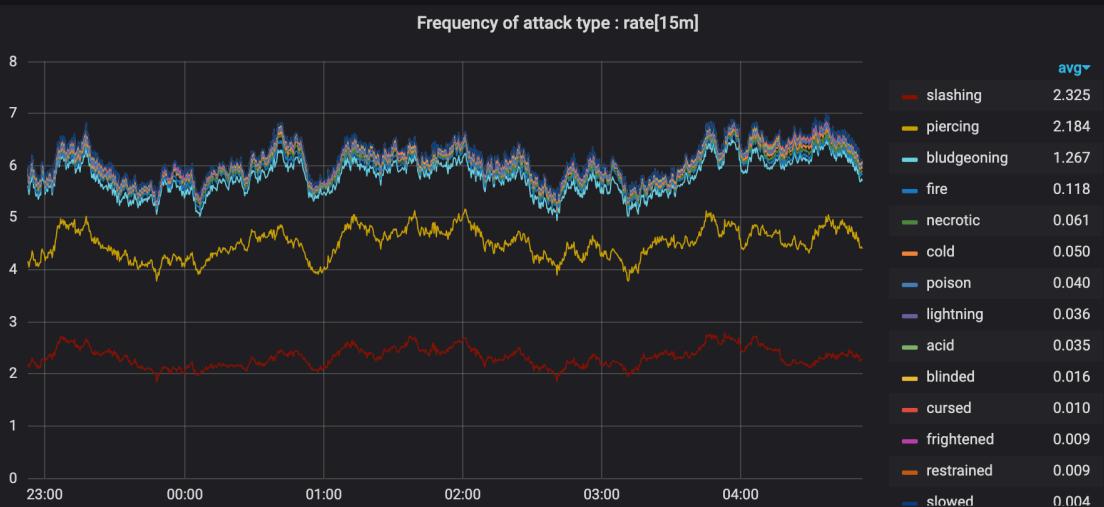
Metric	Total ▾
Overall	438.66 K
hit	257.51 K
miss	131.75 K
critical miss	21.36 K
critical hit	21.18 K
saved hit	6.86 K



Attacks: Damage

```
registry.summary("attacker.damage",
    "attacker", event.getActor().getName(),
    "attackName", event.getName(),
    "attackType", event.getType(),
    "hitOrMiss", event.hitOrMiss())
    .record((double) event.getDamageAmount());
```

- record: instance_attackers:attacker_damage:rate15m
expr: rate(attacker_damage_count[15m])
- record: attackers:attacker_damage:rate15m
expr: sum without(instance, job) (instance_attackers:attacker_damage:rate15m)
- record: instance_damage:attacker_damage:rate15m
expr: rate(attacker_damage_sum[15m])
- record: damage:attacker_damage:rate15m
expr: sum without(instance, job) (instance_damage:attacker_damage:rate15m)



Attacks by creature

Metric	Total ▾
Marilith	6.98 K
Erinyes	5.40 K
Driider	4.77 K
Ancient Gold Dragon	4.50 K
Ancient Brass Dragon	4.43 K
Bone Devil	4.39 K
Ancient Bronze Dragon	4.37 K
Horned Devil	4.32 K
Ice Devil	4.24 K
Ancient White Dragon	4.17 K
Young Silver Dragon	4.15 K
Ancient Blue Dragon	4.14 K
Pit Fiend	4.13 K
Adult Green Dragon	4.08 K
Adult Silver Dragon	4.06 K

1 2 3 4 5 6

7 8 9

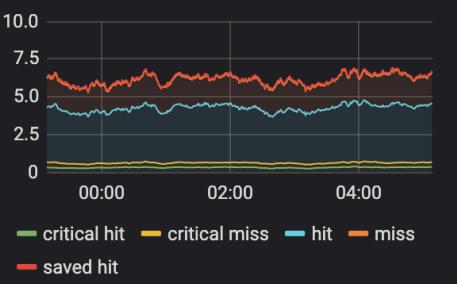
Top 5 Creatures : Average da...

Metric	Total ▾
Remorhaz	32.41
Tarrasque	30.11
Storm Giant	28.15
Iron Golem	24.06
Roc	22.61

Bottom 5 Creatures : Average ...

Metric	Total ▾
Dryad	1.08
Lemure	1.00
Sprite	0.47
Raven	0.44
Homunculus	0.42

Frequency of attack : rate[15m] ▾



Top 5 Attacks: Average Damage

Metric	Total ▾
Lightning Strike	43.05
Lightning Breath	39.51
Acid Breath	32.37
Spit Poison	30.80
Poison Breath	30.57

Bottom 5 Attacks: Average Da...

Metric	Total ▾
Poisoned Dart	2.20
Sling	1.97
Tusk	1.95
Rotting Touch	1.86
Club	1.34

Attacks by weapon

Metric	Total ▾
Bite	100.57 K
Claw	92.92 K
Slam	25.60 K
Claws	20.86 K
Longsword	19.26 K

Tail

Greatsword

Tentacle

Longbow

Scimitar

Greataxe

Hooves

Beak

Fist

Morningstar

1 2 3 4 5 6

7

Average damage per attack : increase[15m]



Encounters and Rounds

How many rounds are there in an encounter?

```
registry.summary("encounter.rounds",
    "numCombatants", label(e.getSize()),
    "targetSelector", e.getSelector(),
    "sizeDelta", label(e.getSizeDelta()))
    .record((double) totalRounds);
```

What are the significant factors that influence that?

- Number of combatants
- How targets are selected (algorithm)
- Creature size

```
# HELP encounter_rounds
# TYPE encounter_rounds summary
encounter_rounds_count{numCombatants="05",sizeDelta="05",targetSelector="HighestHealth",} 18.0
encounter_rounds_sum{numCombatants="05",sizeDelta="05",targetSelector="HighestHealth",} 136.0
encounter_rounds_count{numCombatants="04",sizeDelta="00",targetSelector="LowestHealth",} 7.0
encounter_rounds_sum{numCombatants="04",sizeDelta="00",targetSelector="LowestHealth",} 58.0
encounter_rounds_count{numCombatants="05",sizeDelta="02",targetSelector="LowestHealth",} 86.0
encounter_rounds_sum{numCombatants="05",sizeDelta="02",targetSelector="LowestHealth",} 775.0
encounter_rounds_count{numCombatants="06",sizeDelta="03",targetSelector="SmallestFirst",} 91.0
encounter_rounds_sum{numCombatants="06",sizeDelta="03",targetSelector="SmallestFirst",} 935.0
encounter_rounds_count{numCombatants="03",sizeDelta="00",targetSelector="Random",} 22.0
encounter_rounds_sum{numCombatants="03",sizeDelta="00",targetSelector="Random",} 157.0
encounter_rounds_count{numCombatants="04",sizeDelta="02",targetSelector="Random",} 95.0
encounter_rounds_sum{numCombatants="04",sizeDelta="02",targetSelector="Random",} 654.0
encounter_rounds_count{numCombatants="05",sizeDelta="04",targetSelector="Random",} 35.0
encounter_rounds_sum{numCombatants="05",sizeDelta="04",targetSelector="Random",} 261.0
encounter_rounds_count{numCombatants="05",sizeDelta="01",targetSelector="SmallestFirst",} 42.0
encounter_rounds_sum{numCombatants="05",sizeDelta="01",targetSelector="SmallestFirst",} 398.0
```

▼ Encounters by number of combatants

Encounters : instant	
Metric	Total
Overall	14.04 K
03 combatants	2.80 K
04 combatants	2.77 K
05 combatants	2.91 K
06 combatants	2.78 K
02 combatants	2.79 K

Encounters by number of combatants : increase[15m]



▼ Rounds / Encounter by number of combatants

Rounds : instant	
Metric	Total
Overall	102.71 K
06 combatants	29.68 K
02 combatants	9.66 K
03 combatants	16.04 K
04 combatants	20.88 K
05 combatants	26.45 K

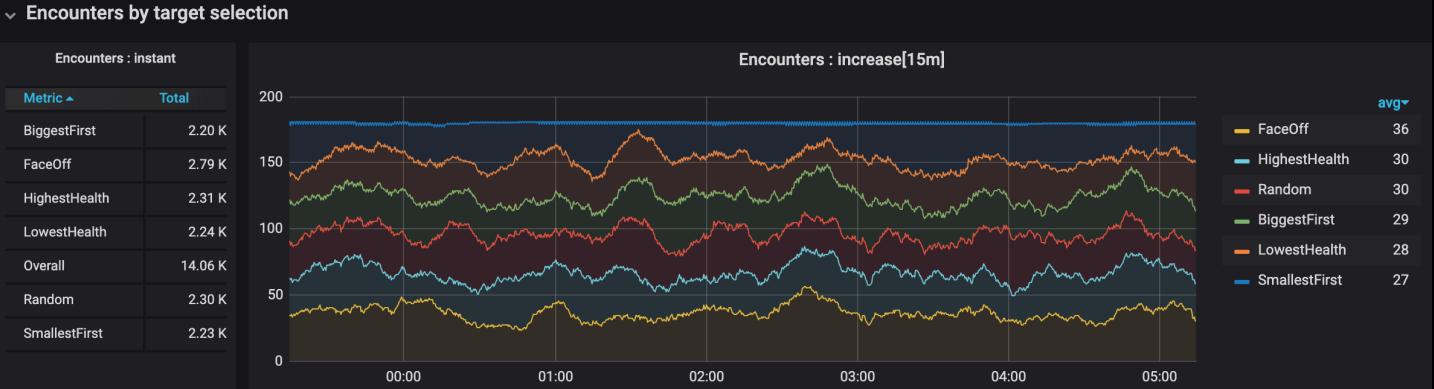
Rounds by number of combatants : increase[15m]



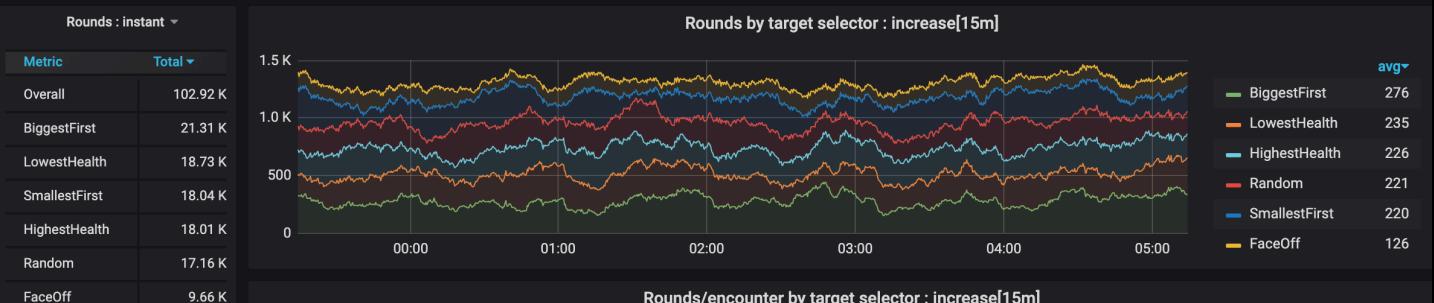
Rounds / Encounter by number of combatants : increase[15m]



Target Selection



▼ Rounds / Encounter by target selection



Rounds/encounter by target selector : increase[15m]



Target Selection

4 Combatants

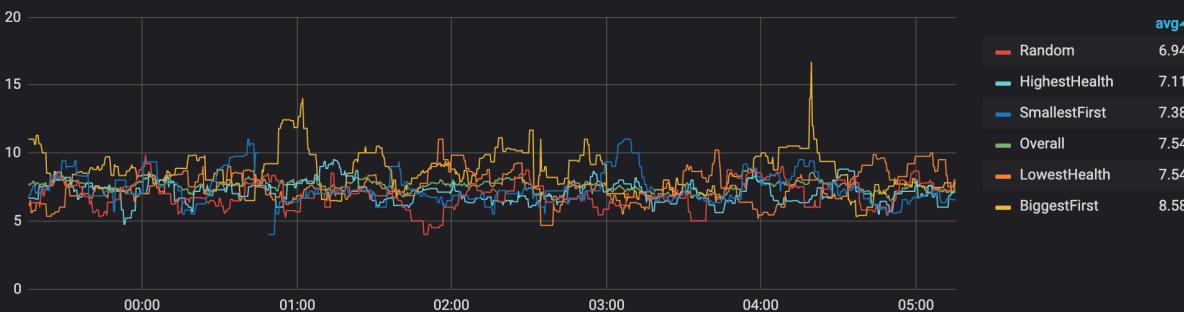
▼ Encounters by target selector



▼ Rounds / encounter by target selector



Average rounds/encounter by sizeDelta : increase[15m]

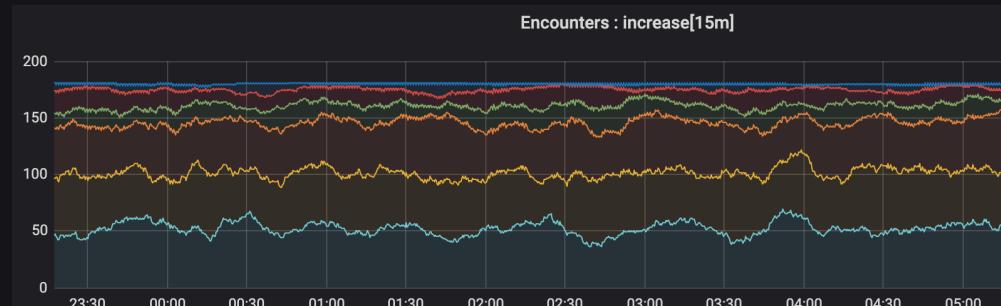


Size difference

▼ Encounters by difference in size

Encounters : instant

Metric ▲	Total
Overall	14.10 K
sizeDelta 00	1.16 K
sizeDelta 01	3.92 K
sizeDelta 02	4.05 K
sizeDelta 03	3.43 K
sizeDelta 04	1.09 K
sizeDelta 05	441.00

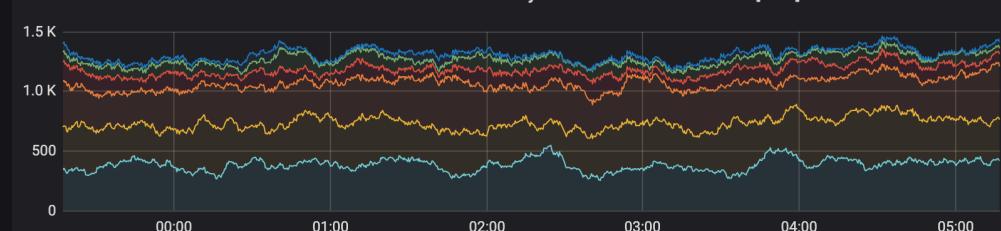


▼ Rounds / Encounter by difference in size

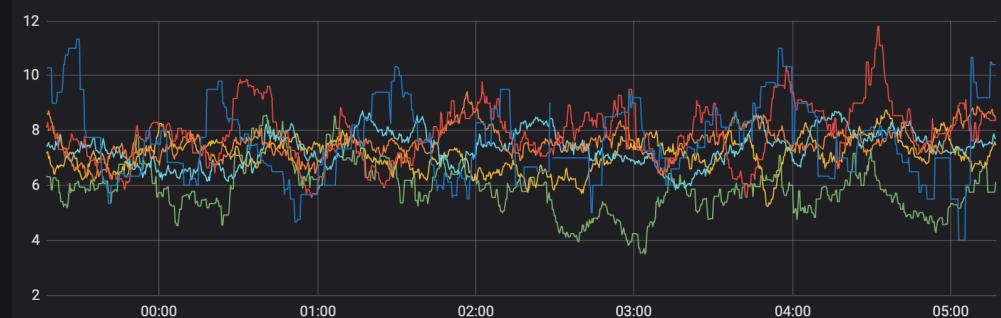
Rounds : instant

Metric	Total ▾
Overall	103.17 K
sizeDelta 02	30.16 K
sizeDelta 01	27.67 K
sizeDelta 03	26.57 K
sizeDelta 04	8.44 K
sizeDelta 00	6.79 K
sizeDelta 05	3.55 K

Rounds by difference in size : increase[15m]



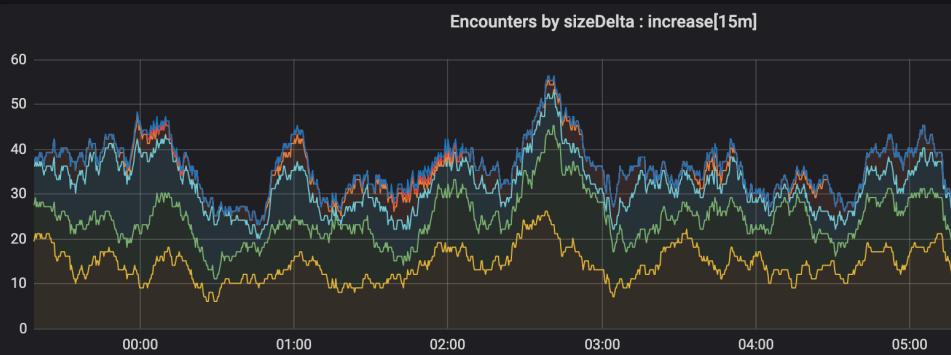
Rounds/encounter by difference in size : increase[15m]



Size difference 2 Combatants

▼ Encounters by difference in size

Encounters : instant	
Metric ▲	Total
Overall	2.79 K
sizeDelta 00	778.00
sizeDelta 01	1.13 K
sizeDelta 02	582.00
sizeDelta 03	258.00
sizeDelta 04	40.00
sizeDelta 05	9.00

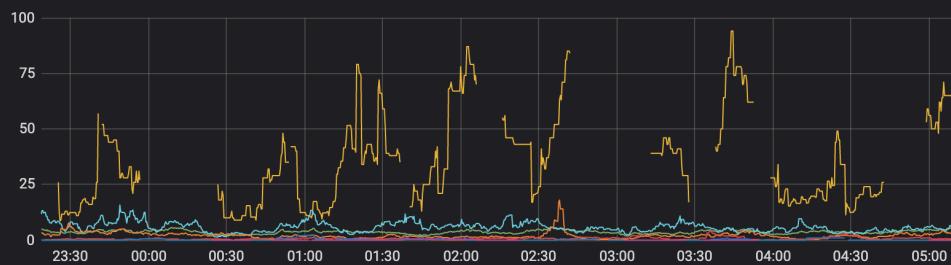


▼ Rounds / Encounter by difference in size

Rounds : instant	
Metric ▲	Total
Overall	9.70 K
sizeDelta 00	3.64 K
sizeDelta 01	4.12 K
sizeDelta 02	1.44 K
sizeDelta 03	450.00
sizeDelta 04	43.00
sizeDelta 05	9.00



Average rounds/encounter by sizeDelta : increase[30m]

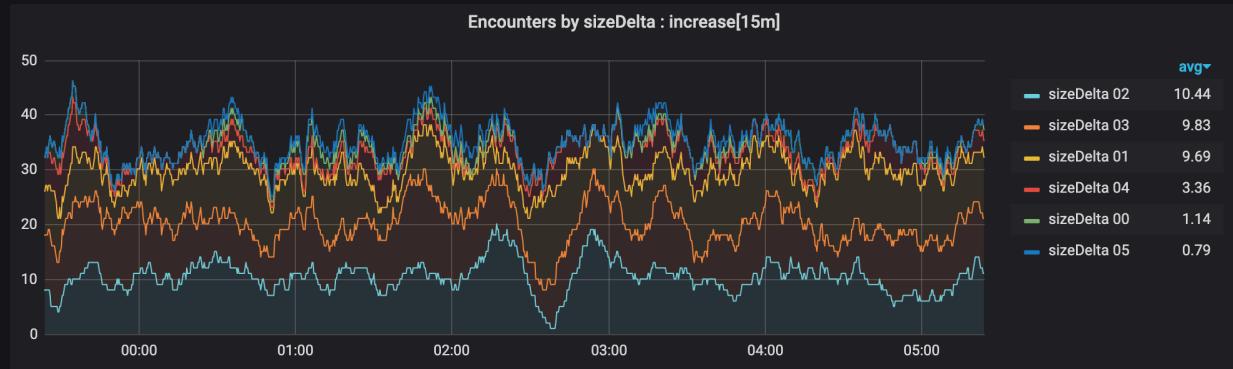


Size difference 4 Combatants

▼ Encounters by difference in size

Encounters : instant

Metric ▲	Total
Overall	2.80 K
sizeDelta 00	93.00
sizeDelta 01	798.00
sizeDelta 02	890.00
sizeDelta 03	720.00
sizeDelta 04	230.00
sizeDelta 05	69.00



▼ Rounds / Encounter by difference in size

Rounds : instant

Metric ▲	Total
Overall	21.10 K
sizeDelta 00	898.00
sizeDelta 01	6.61 K
sizeDelta 02	6.72 K
sizeDelta 03	4.96 K
sizeDelta 04	1.45 K
sizeDelta 05	455.00



Average rounds/encounter by sizeDelta : increase[15m]



Custom metrics for the win

- Identify bugs from corner cases
- Observe impact of implementation choices
- Validate assumptions

Meter types: MicroProfile

Some inconsistencies with Prometheus: calculation in the app, rather than exposing raw data

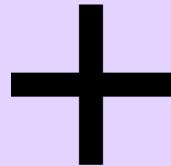
Incrementing value

Observed value

Distribution of values (buckets)

Bucket x Timer

Counter



Gauge



Histogram



Summary



Counter

Can be
decremented, too

Gauge

ConcurrentGauge

Histogram

Count
Mean, max, min, stddev: Gauge
No sum, only value distribution

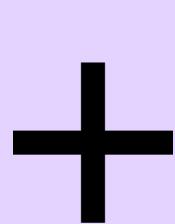
Timer

SimpleTimer

Duration (total time)
AND count / time
as Histogram

Meter types: OpenTelemetry

Incrementing value



Observed value



Distribution of values (buckets)



Bucket x Timer

Summary



Counter

Rise and fall
Only a sum

LastValue

Measure

Measure

Sum, Count, Min, Max

Thank you!

@ebullientworks

<https://github.com/ebullient/monster-combat>

