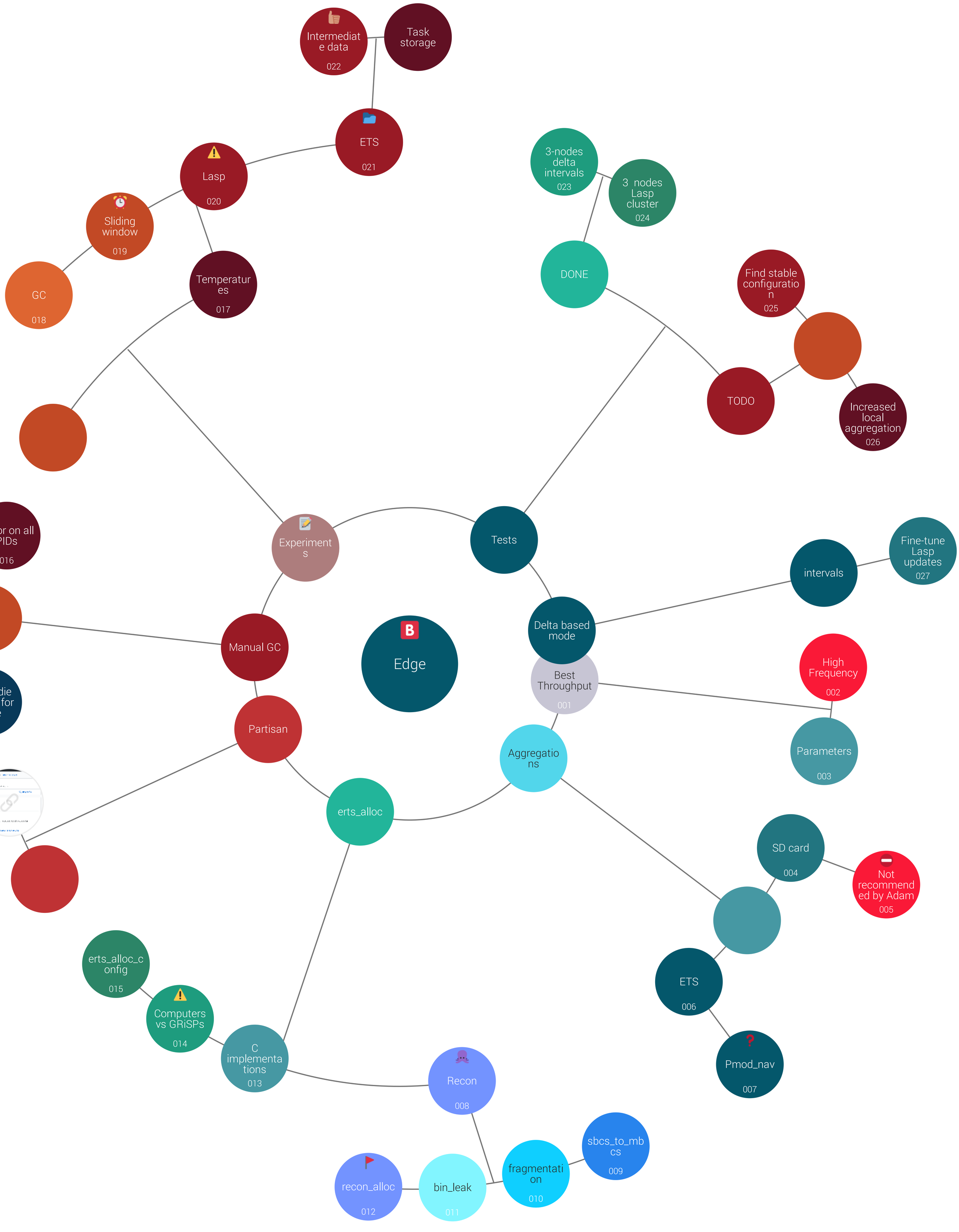
Oct 29, 2018 01:29

Edge



001 Best Throughput

Ops x Size

Interval

002 High Frequency

Short delays combined with high (10-30sec) delta gossip intervals overload memory

003 Parameters

pressure :

(Size × Count) / Interval of updates

004 SD card

Could a local storage of aggregates on SD card i.e. persistent allow for less memory and network load?

005 Not recommended by Adam

Not durable

006 ETS

Hypothesis :

Could a local ets storage of aggregates instead of process heap storage improve memory usage while preserving sufficient resilience?

Less Lasp updates but also les traffic and process memory usage.

007 Pmod_nav

aggregate temperatures in ETS table with erlang:monotonic_time(),temp k/v pairs

008 Recon

@ferd's application for inspection, mostly recon:alloc

009 sbcs_to_mbcs

ratio has improved

010 fragmentation

acceptor_pool noticed in fragmented procs

011 bin_leak

Indicates leaking binaries are close to none

012 recon_alloc

Points out high unused allocated memory

:memory(unused,max)

013 C implementations

available on OTP repo, show detailed specs and implementation of allocators setup

014 Computers vs GRiSPs

mseg_alloc process :

growth stages = 10
lmbcs = 1024*1024
smbcs = 1024

Sizes of multiblock carriers allocated through mseg_allocare decided based on the following parameters:

The values of the largest multiblock carrier size (lmbcs)

The smallest multiblock carrier size (smbcs)

The multiblock carrier growth stages (mbcgs)

If nc is the current number of multiblock carriers (the main multiblock carrier excluded) managed by an allocator, the size of the next mseg_alloc multiblock carrier allocated by this allocator is roughly smbcs+nc*(lmbcs-smbcs)/mbcgs when nc <= mbcgs, and lmbcs when nc > mbcgs. If the value of parameter sbct is larger than the value of parameter lmbcs, the allocator may have to create multiblock carriers that are larger than the value of parameter lmbcs, though. Singleblock carriers allocated through mseg_alloc are sized to whole pages.

let nc <= mbcgs :

next mbc size = smbcs + nc × (lmbcs - smbcs)

= lmbcs otherwise

12 cores -> 13 eheap allocs
1 core -> 2 allocs
---> SMP enabled devices

015 erts_alloc_config

a custom erts_alloc config file can be generated based on learning from scenario runs. It can possibly optimize alloc configuration for more specific applications.

016 Major on all PIDs

up to 2MB collected : increases lifetime

017 Temperatures

pmod_nav

018 GC

Timed GC's included

019 Sliding window

Easily maintained with ETS and monotonic_time() for each table

020 Lasp

update crdt periodically or on-demand only?

021 ETS

storing measures in local ETS tables and performing periodical updates of means. {erlang :monotonic_time() , [Temp]} tuples

022 Intermediate data

tasks could trigger propagation based on table content

023 3-nodes delta intervals

1---2---3

Node 2 crashes first even if more traffic generated at node 3 => relays could require additional backpressure

delta interval for Lasp set to 10,20 and 30 seconds

Similar traffic on all nodes

cycle : 1000 iterations
add 100bits
wait 500ms
rmv 100bits
wait 500ms

No memory improvement

024 3 nodes Lasp cluster

1---2---3

Node 2 crashes first even if more traffic generated at node 3 => relays could require additional backpressure

Max throughput : node3

cycle : 1000 iterations
add 200bits
wait 500ms
rmv 200bits
wait 500ms

025 Find stable configuration

Determine maximum traffic enabling long lived nodes

026 Increased local aggregation

Same operations except only 1 in N iterations propagate data in Lasp

027 Fine-tune Lasp updates params

Packet config :

LD = Low Delay
LS = Low Size
LC = Low Count

1. LD and !LS and !LC
2. LD and LS and !LC
3. all
4. !LD and LS and !LC
5. LD and !LS and !LC