GigA+ (Gigaplus)

Architectural Overview (beta1)

GigA+ delivers streams over HTTP and HLS, with DVR support (for HLS)

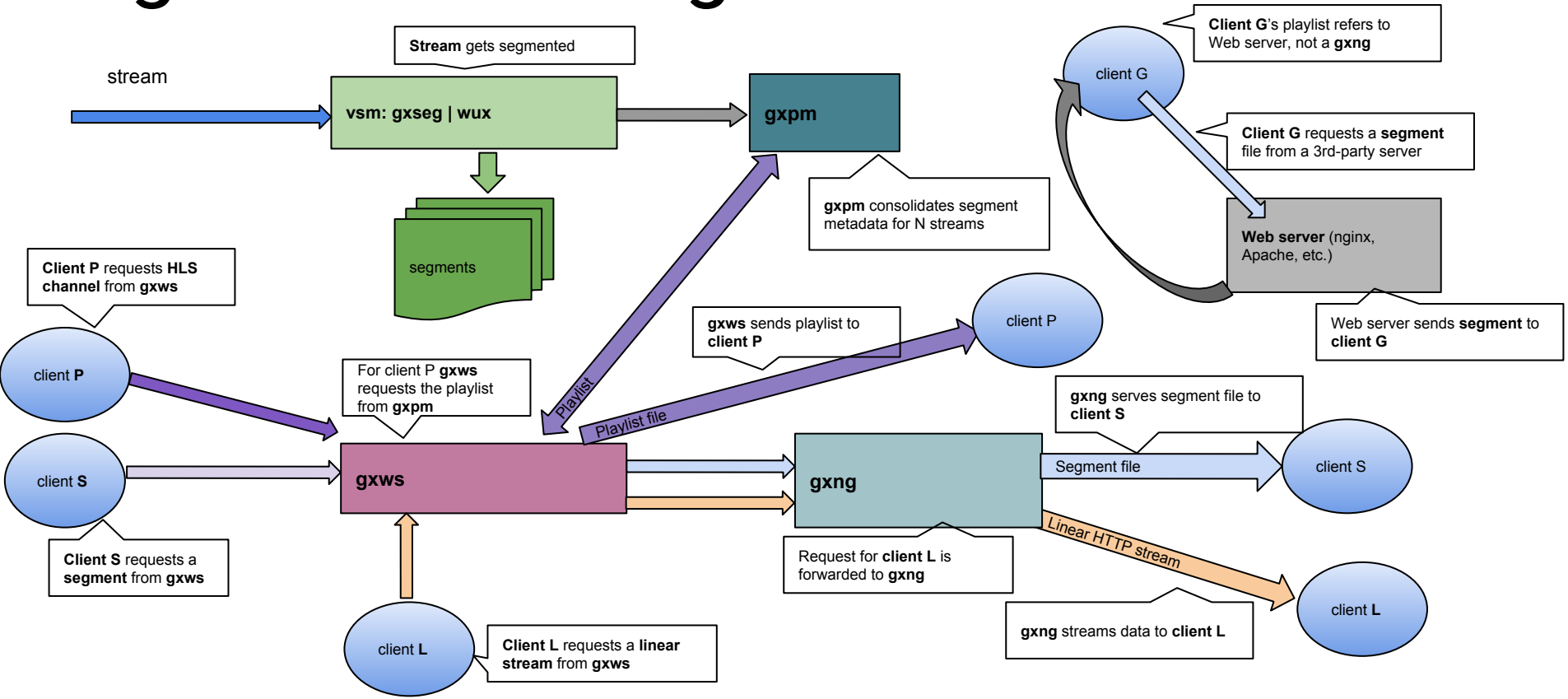
What is in GigA+?

- **Full Gigapxy 1.x functionality** - delivering linear content;
- **HLS +DVR support** - live and time-shifted content via HLS;
- **Load-balancing**: HLS requests spread across multiple servers.

Core modules:

- **gxws** (web server) - handles client (HTTP) requests;
- **gxng** (engine) - handles delivery of content; for HLS could be replaced by a 3-rd party server;
- **gxseg** (segmentation engine) - splits linear streams into segments;
- **gxpm** (playlist manager) - processes segment metadata, serves HLS playlists;
- **wux** (message proxy) - facilitates message passing from gxseg to the playlist manager;
- **vsm** (stream manager) - prepares streams for delivery via HLS;
- **dwg** (download agent) - downloads segments from remote hosts;
- **flbr** (request broker) - distributes requests across servers.

GigA+ data flow (single-host)



Module interaction (single-host)

Linear streams: **gxws** gets and parses the (HTTP) request, **gxng** serves the stream.

HLS streams get prepared by **vsm**.
Each channel/stream has its own **vsm** instance.
vsm uses **gxseg** to create segments.
gxseg passes metadata to **gxpm** via wux.
gxpm serves playlists to **gxws**.

gxws receives a playlist request.
If playlist is cached, it serves it to client, otherwise
gxws requests playlist from **gxpm**.
Playlist contains segment URLs that
can go back to gxws or refer to (3rd-party) web server.

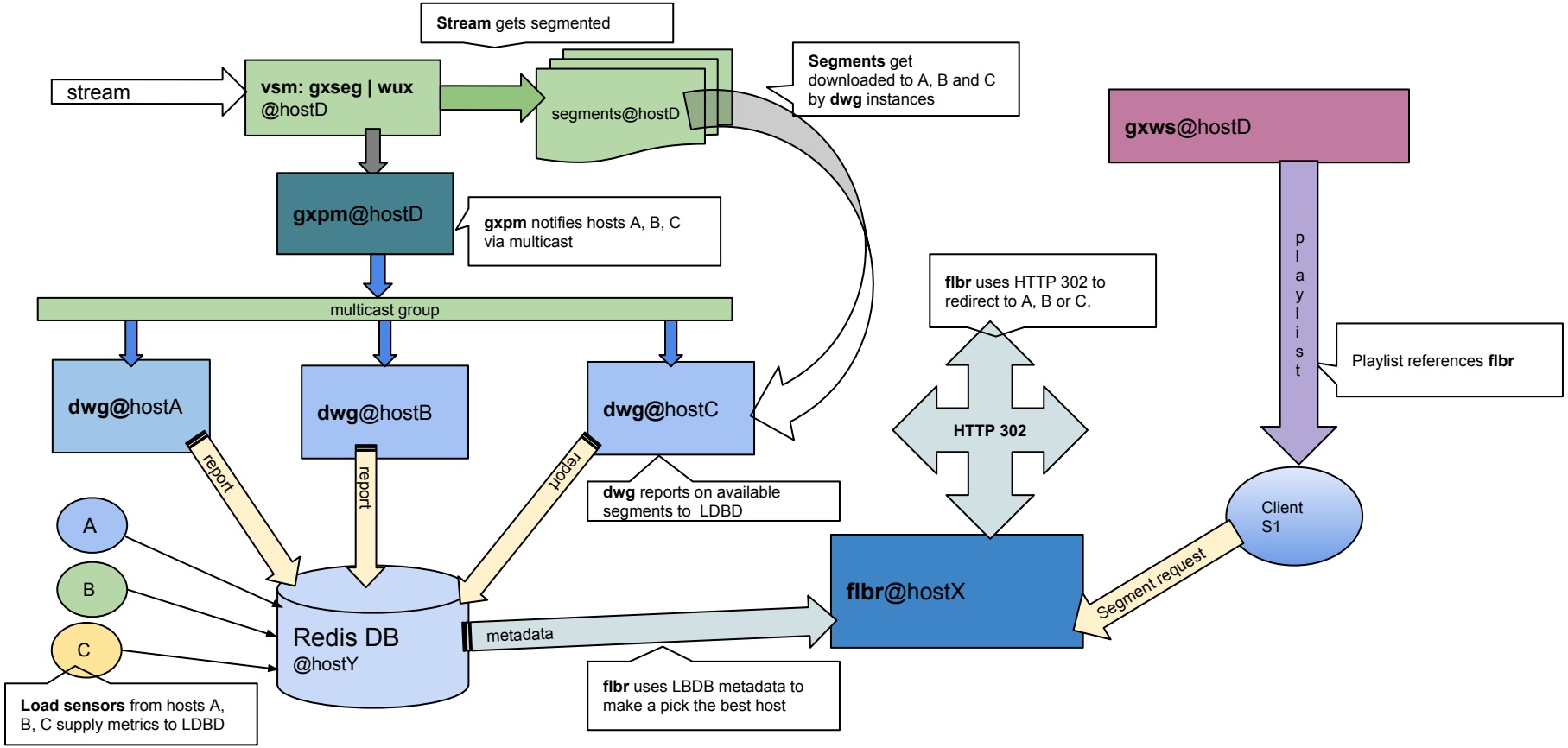
vsm makes sure that data is segmented 24/7.
gxpm keeps track of all segments within the storage (DVR) capacity of the channel.
gxpm creates playlists for *live* or *DVR* requests
wux is a pass-through from **gxseg** to **gxpm**.

gxws receives a segment request,
It passes the request to gxng.
gxng serves the request (sends segment to client).

Web server receives a segment request,
It serves the request (sends segment to client).

Segments can be **replicated** and served from **multiple hosts**; requests **load-balanced** between hosts.

GigA+ data flow (multi-host)



Module interaction (multi-host)

Playlist URLs reference *Web server* where **flbr** runs as a FastCGI plug-in. **flbr** can run on any host.

gxpm@origin sends segment info to multicast group **M** to notify *data hosts* of new segments.

Data hosts (1..N) hold replicas of data segments.
dwg@K subscribes to multicast group **M**.
dwg@K downloads segments from the *origin* to host K.
dwg reports every download to **LBDB**.

Load sensors provide load metrics from *data hosts*.
Sensors update **LBDB**.

Load-balancing database - **LBDB** (Redis) consolidates metadata from participating hosts.

Web server (hostX) receives a request (for segment), recognizes it as load-balanced kind and passes it to **flbr**.

flbr@X queries **LDBD** and decides which data host must be used. It then issues **HTTP 302** referencing that host.

flbr@X can use sensor metrics to decide which data host to pick.

Software considerations

GigA+ has many more modules, compared to *Gigapxy 1.x* (has only two modules). Documentation must be read, starting with *man* ***gigaplus***. Get one-for-all *PDF document* from the *website*.

Installation is trivial, but configuration is NOT. Need to read a page on the module before configuring one. Start with *man* ***gxws*** (after reading *man* ***gigaplus***).

Mind the ***core dumps***. You'll need them to report bugs, make sure they are *enabled* for all running modules. Set up a crash-handling system to grab *core dump(s)* + relevant *logs*.

Pick a **fast file system** for segment storage (many small files). Research, run comparison tests. Viable candidates: xfs, ext4, UFS (FreeBSD), zfs.

Disable ***gxng buffers*** if you're serving **HLS** only, not linear streams. Do not waste your RAM. Check if you'd rather have *gxng* or *nginx* serve your data segments.

HOWTO manuals are provided, not just *man* pages.

man ***gxa-setup*** - read it first.
man ***gxa-lb-setup*** - if scaling to multiple servers.

Log space is important, make sure there's plenty. Put your logs in a dedicated (very *fast*) partition/disk.

Make it work on a ***single host*** before scaling to a ***multiple-host*** configuration. K.I.S.S.

Tuning up system (**sysctl**) parameters is crucial. But you know that already.

Hardware considerations

You **don't** need the *latest* and the *greatest* H/W, but you surely need *fast I/O* and plenty of storage (for the segments, if doing HLS+DVR).

Spread the load to CPU(s) and cores. One CPU with many cores may work better than 2 CPUs with network-intensive I/O.

Example (used with *Gigapxy 1.x*):

```
CPU: 1 x Intel(R) Xeon(R) CPU   X5355  @ 2.66GHz
Memory: 64Gb (8X8Gb)
Network card:  1 x Intel x520 10-Gigabit Dual Ports
SSD: 2 x Intel 530 series 120Gb (RAID 1+0) (for system)
SSD: 1 x Kingston 120Gb (for buffers files)
```

NB: *storage requirements* depend on the # of channels and their *DVR* parameters.

Segmenting process (via *gxseg*) stress on the **CPU**. It also constantly uses the file system and the underlying **disks**.

Data hosts **don't** need a fast CPU, but require fast **disks** and **NIC**(s).

LDBD host requires maximum availability (**min CPU load**) and **fast I/O (disk, NIC**s).

Si vis pacem, para bellum.

Plan for **redundancy**. A ***rollover server*** (for the most crucial part) could be better than RAID.

RAM will be needed by segmenters. Plenty of RAM gives OS more **filesystem cache** to make I/O faster.

What's coming next?

It depends a lot on your feedback, seriously. Your requests truly matter and are listened to.

IGMPv3 SSM support for *dwg* URLs.

Documentation and HOWTOs translated to other languages.

External (enabled by grafana) application reports to replace *gxws*-based TPS reports.

DRM, MPEG4 segments, DASH

Contacts: team@gigapxy.com, pcherenkov@gigapxy.com