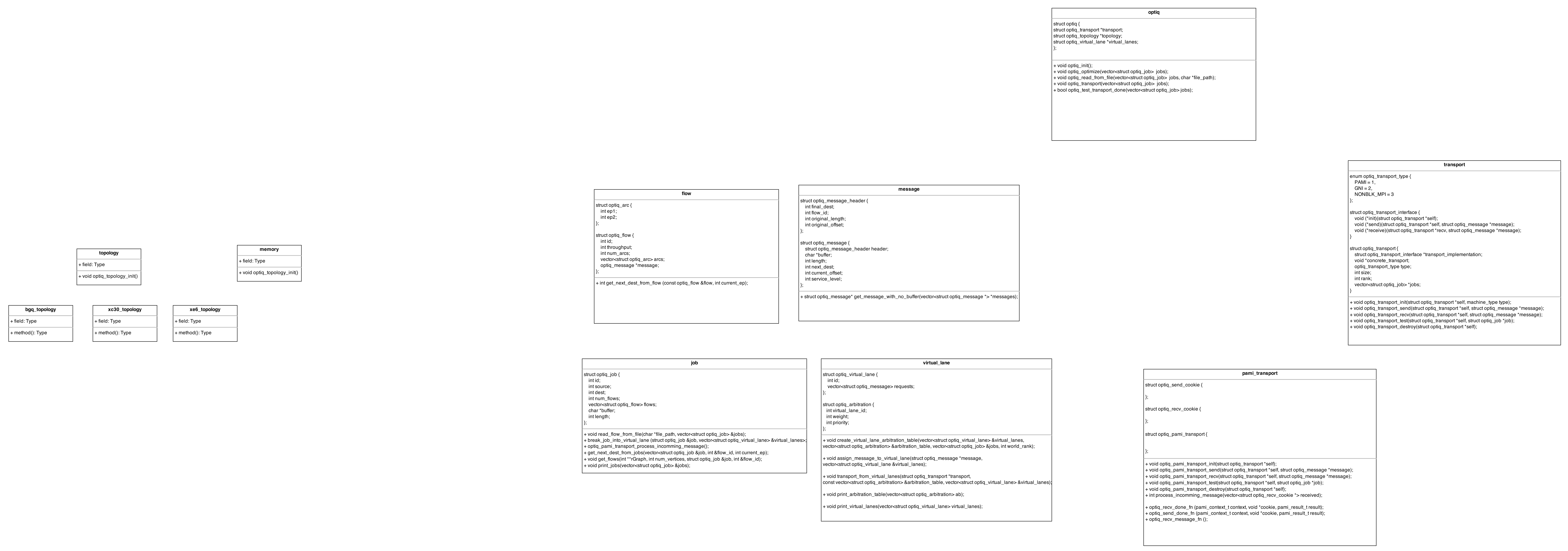

optiq

```
struct optiq {
struct optiq_transport *transport;
struct optiq_topology *topology;
struct optiq_virtual_lane *virtual_lanes;
};
```

```
+ void optiq_init();
+ void optiq_optimize(vector<struct optiq_job>  jobs);
+ void optiq_read_from_file(vector<struct optiq_job>  jobs, char *file_path);
+ void optiq_transport(vector<struct optiq_job>  jobs);
+ bool optiq_test_transport_done(vector<struct optiq_job> jobs);
```

transport

```
enum optiq_transport_type {
   PAMI = 1,
   GNI = 2,
   NONBLK_MPI = 3
};

struct optiq_transport_interface {
   void (*init)(struct optiq_transport *self);
   void (*send)(struct optiq_transport *self, struct optiq_message *message);
   void (*receive)(struct optiq_transport *recv, struct optiq_message *message);
}

struct optiq_transport {
   struct optiq_transport_interface *transport_implementation;
   void *concrete_transport;
   optiq_transport_type type;
   int size;
   int rank;
   vector<struct optiq_job> *jobs;
}
```

```
+ void optiq_transport_init(struct optiq_transport *self, machine_type type);
+ void optiq_transport_send(struct optiq_transport *self, struct optiq_message *message);
+ void optiq_transport_recv(struct optiq_transport *self, struct optiq_message *message);
+ void optiq_transport_test(struct optiq_transport *self, struct optiq_job *job);
+ void optiq_transport_destroy(struct optiq_transport *self);
```

flow

```
struct optiq_arc {
   int ep1;
   int ep2;
};

struct optiq_flow {
   int id;
   int throughput;
   int num_arcs;
   vector<struct optiq_arc> arcs;
   optiq_message *message;
};
```

```
+ int get_next_dest_from_flow (const optiq_flow &flow, int current_ep);
```

message

```
struct optiq_message_header {
   int final_dest;
   int flow_id;
   int original_length;
   int original_offset;
};

struct optiq_message {
   struct optiq_message_header header;
   char *buffer;
   int length;
   int next_dest;
   int current_offset;
   int service_level;
};
```

```
+ struct optiq_message* get_message_with_no_buffer(vector<struct optiq_message *> *messages);
```

topology

+ field: Type

+ void optiq_topology_init()

memory

+ field: Type

+ void optiq_topology_init()

bgq_topology

+ field: Type

+ method(): Type

xc30_topology

+ field: Type

+ method(): Type

xe6_topology

+ field: Type

+ method(): Type

job

```
struct optiq_job {
   int id;
   int source;
   int dest;
   int num_flows;
   vector<struct optiq_flow> flows;
   char *buffer;
   int length;
};
```

```
+ void read_flow_from_file(char *file_path, vector<struct optiq_job> &jobs);
+ break_job_into_virtual_lane (struct optiq_job &job, vector<struct optiq_virtual_lane> &virtual_lanes>;
+ optiq_pami_transport_process_incomming_message();
+ get_next_dest_from_jobs(vector<struct optiq_job &job, int &flow_id, int current_ep);
+ void get_flows(int **rGraph, int num_vertices, struct optiq_job &job, int &flow_id);
+ void print_jobs(vector<struct optiq_job> &jobs);
```

virtual_lane

```
struct optiq_virtual_lane {
    int id;
    vector<struct optiq_message> requests;
};

struct optiq_arbitration {
   int virtual_lane_id;
   int weight;
   int priority;
};
```

```
+ void create_virtual_lane_arbitration_table(vector<struct optiq_virtual_lane> &virtual_lanes,
vector<struct optiq_arbitration> &arbitration_table, vector<struct optiq_job> &jobs, int world_rank);

+ void assign_message_to_virtual_lane(struct optiq_message *message,
vector<struct optiq_virtual_lane &virtual_lanes);

+ void transport_from_virtual_lanes(struct optiq_transport *transport,
const vector<struct optiq_arbitration> &arbitration_table, vector<struct optiq_virtual_lane> &virtual_lanes);

+ void print_arbitration_table(vector<struct optiq_arbitration> ab);

+ void print_virtual_lanes(vector<struct optiq_virtual_lane> virtual_lanes);
```

pami_transport

```
struct optiq_send_cookie {

};

struct optiq_recv_cookie {

};

struct optiq_pami_transport {


};
```

```
+ void optiq_pami_transport_init(struct optiq_transport *self);
+ void optiq_pami_transport_send(struct optiq_transport *self, struct optiq_message *message);
+ void optiq_pami_transport_recv(struct optiq_transport *self, struct optiq_message *message);
+ void optiq_pami_transport_test(struct optiq_transport *self, struct optiq_job *job);
+ void optiq_pami_transport_destroy(struct optiq_transport *self);
+ int process_incomming_message(vector<struct optiq_recv_cookie *> received);

+ optiq_recv_done_fn (pami_context_t context, void *cookie, pami_result_t result);
+ optiq_send_done_fn (pami_context_t context, void *cookie, pami_result_t result);
+ optiq_recv_message_fn ();
```