Complete Class Diagram

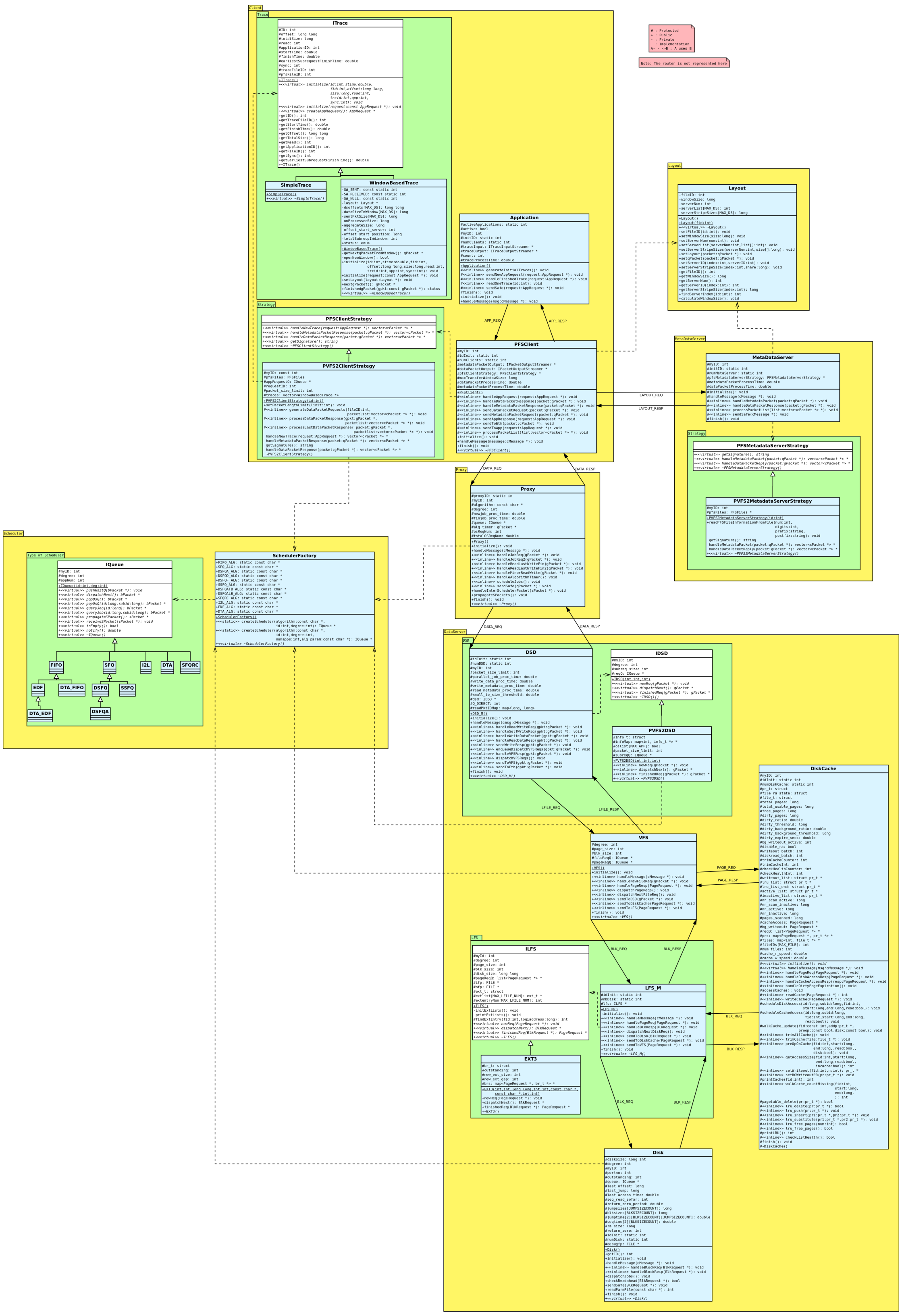


Figure 1: Class Diagram PFSSim

Class Diagram Client Package

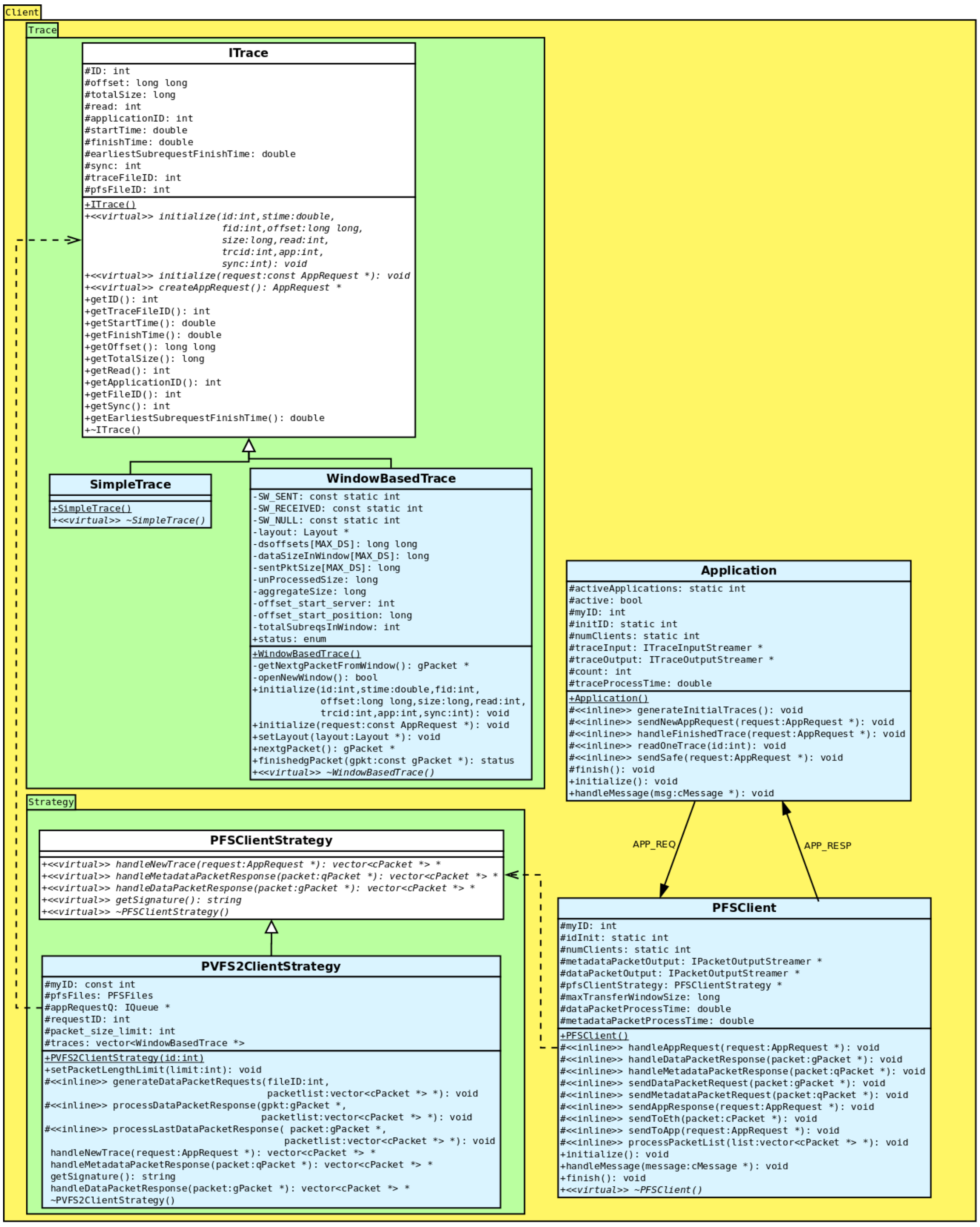
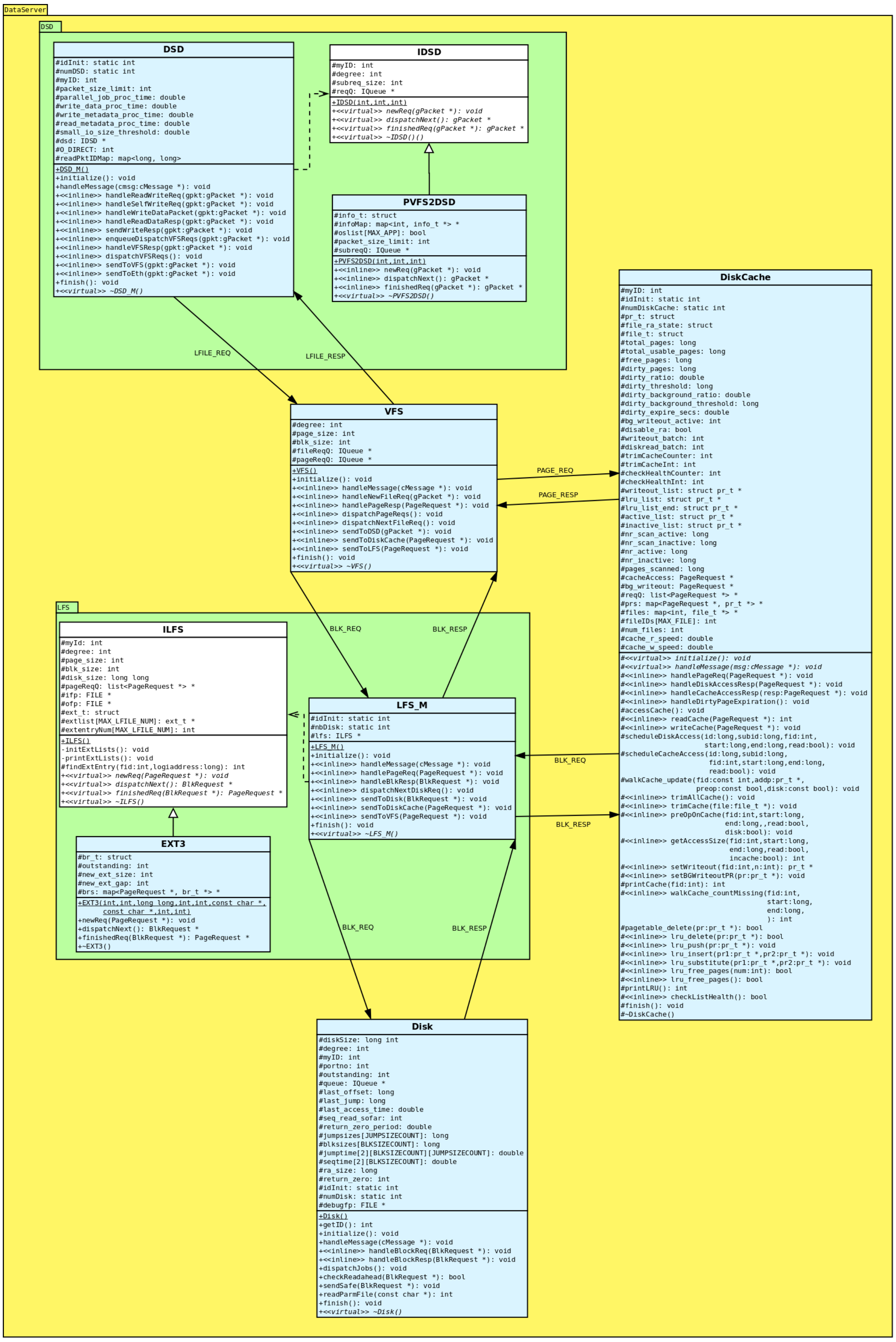

Client
Trace
ITrace
#ID: int
#offset: long long
#totalSize: long
#read: int
#applicationID: int
#startTime: double
#finishTime: double
#earliestSubrequestFinishTime: double
#sync: int
#traceFileID: int
#pfsFileID: int
+ITrace()
+<<virtual>> initialize(id:int,stime:double,
fid:int,offset:long long,
size:long,read:int,
trcid:int,app:int,
sync:int): void
+<<virtual>> initialize(request:const AppRequest *): void
+<<virtual>> createAppRequest(): AppRequest *
+getID(): int
+getTraceFileID(): int
+getStartTime(): double
+getFinishTime(): double
+getOffset(): long long
+getTotalSize(): long
+getRead(): int
+getApplicationID(): int
+getFileID(): int
+getSync(): int
+getEarliestSubrequestFinishTime(): double
+~ITrace()
SimpleTrace
+SimpleTrace()
+<<virtual>> ~SimpleTrace()
WindowBasedTrace
-SW_SENT: const static int
-SW_RECEIVED: const static int
-SW_NULL: const static int
-layout: Layout *
-dsoffsets[MAX_DS]: long long
-dataSizeInWindow[MAX_DS]: long
-sentPktSize[MAX_DS]: long
-unProcessedSize: long
-aggregateSize: long
-offset_start_server: int
-offset_start_position: long
-totalSubreqsInWindow: int
+status: enum
+WindowBasedTrace()
-getNextgPacketFromWindow(): gPacket *
-openNewWindow(): bool
+initialize(id:int,stime:double,fid:int,
offset:long long,size:long,read:int,
trcid:int,app:int,sync:int): void
+initialize(request:const AppRequest *): void
+setLayout(layout:Layout *): void
+nextgPacket(): gPacket *
+finishedgPacket(gpkt:const gPacket *): status
+<<virtual>> ~WindowBasedTrace()
Application
#activeApplications: static int
#active: bool
#myID: int
#initID: static int
#numClients: static int
#traceInput: ITraceInputStreamer *
#traceOutput: ITraceOutputStreamer *
#count: int
#traceProcessTime: double
+Application()
#<<inline>> generateInitialTraces(): void
#<<inline>> sendNewAppRequest(request:AppRequest *): void
#<<inline>> handleFinishedTrace(request:AppRequest *): void
#<<inline>> readOneTrace(id:int): void
#<<inline>> sendSafe(request:AppRequest *): void
#finish(): void
+initialize(): void
+handleMessage(msg:cMessage *): void
APP_REQ
APP_RESP
Strategy
PFSClientStrategy
+<<virtual>> handleNewTrace(request:AppRequest *): vector<cPacket *> *
+<<virtual>> handleMetadataPacketResponse(packet:qPacket *): vector<cPacket *> *
+<<virtual>> handleDataPacketResponse(packet:gPacket *): vector<cPacket *> *
+<<virtual>> getSignature(): string
+<<virtual>> ~PFSClientStrategy()
PVFS2ClientStrategy
#myID: const int
#pfsFiles: PFSFiles
#appRequestQ: IQueue *
#requestID: int
#packet_size_limit: int
#traces: vector<WindowBasedTrace *>
+PVFS2ClientStrategy(id:int)
+setPacketLengthLimit(limit:int): void
#<<inline>> generateDataPacketRequests(fileID:int,
packetlist:vector<cPacket *> *): void
#<<inline>> processDataPacketResponse(gpkt:gPacket *,
packetlist:vector<cPacket *> *): void
#<<inline>> processLastDataPacketResponse(packet:gPacket *,
packetlist:vector<cPacket *> *): void
handleNewTrace(request:AppRequest *): vector<cPacket *> *
handleMetadataPacketResponse(packet:qPacket *): vector<cPacket *> *
getSignature(): string
handleDataPacketResponse(packet:gPacket *): vector<cPacket *> *
~PVFS2ClientStrategy()
PFSClient
#myID: int
#idInit: static int
#numClients: static int
#metadataPacketOutput: IPacketOutputStreamer *
#dataPacketOutput: IPacketOutputStreamer *
#pfsClientStrategy: PFSClientStrategy *
#maxTransferWindowSize: long
#dataPacketProcessTime: double
#metadataPacketProcessTime: double
+PFSClient()
#<<inline>> handleAppRequest(request:AppRequest *): void
#<<inline>> handleDataPacketResponse(packet:gPacket *): void
#<<inline>> handleMetadataPacketResponse(packet:qPacket *): void
#<<inline>> sendDataPacketRequest(packet:gPacket *): void
#<<inline>> sendMetadataPacketRequest(packet:qPacket *): void
#<<inline>> sendAppResponse(request:AppRequest *): void
#<<inline>> sendToEth(packet:cPacket *): void
#<<inline>> sendToApp(request:AppRequest *): void
#<<inline>> processPacketList(list:vector<cPacket *> *): void
+initialize(): void
+handleMessage(message:cMessage *): void
+finish(): void
+<<virtual>> ~PFSClient()

Figure 2: Class Diagram PFSSim Package Client

Class Diagram Data Server Package



DataServer
DSD
DSD
#idInit: static int
#numDSD: static int
#myID: int
#packet_size_limit: int
#parallel_job_proc_time: double
#write_data_proc_time: double
#write_metadata_proc_time: double
#read_metadata_proc_time: double
#small_io_size_threshold: double
#dsd: IDSD *
#O_DIRECT: int
#readPktIDMap: map<long, long>
+DSD_M()
+initialize(): void
+handleMessage(cmsg:cMessage *): void
+<<inline>> handleReadWriteReq(gpkt:gPacket *): void
+<<inline>> handleSelfWriteReq(gpkt:gPacket *): void
+<<inline>> handleWriteDataPacket(gpkt:gPacket *): void
+<<inline>> handleReadDataResp(gpkt:gPacket *): void
+<<inline>> sendWriteResp(gpkt:gPacket *): void
+<<inline>> enqueueDispatchVFSReqs(gpkt:gPacket *): void
+<<inline>> handleVFSResp(gpkt:gPacket *): void
+<<inline>> dispatchVFSReqs(): void
+<<inline>> sendToVFS(gpkt:gPacket *): void
+<<inline>> sendToEth(gpkt:gPacket *): void
+finish(): void
+<<virtual>> ~DSD_M()
IDSD
#myID: int
#degree: int
#subreq_size: int
#reqQ: IQueue *
+IDSD(int,int,int)
+<<virtual>> newReq(gPacket *): void
+<<virtual>> dispatchNext(): gPacket *
+<<virtual>> finishedReq(gPacket *): gPacket *
+<<virtual>> ~IDSD()()
PVFS2DSD
#info_t: struct
#infoMap: map<int, info_t *> *
#oslist[MAX_APP]: bool
#packet_size_limit: int
#subreqQ: IQueue *
+PVFS2DSD(int,int,int)
+<<inline>> newReq(gPacket *): void
+<<inline>> dispatchNext(): gPacket *
+<<inline>> finishedReq(gPacket *): gPacket *
+<<virtual>> ~PVFS2DSD()
LFILE_REQ
LFILE_RESP
VFS
#degree: int
#page_size: int
#blk_size: int
#fileReqQ: IQueue *
#pageReqQ: IQueue *
+VFS()
+initialize(): void
+<<inline>> handleMessage(cMessage *): void
+<<inline>> handleNewFileReq(gPacket *): void
+<<inline>> handlePageResp(PageRequest *): void
+<<inline>> dispatchPageReqs(): void
+<<inline>> dispatchNextFileReq(): void
+<<inline>> sendToDSD(gPacket *): void
+<<inline>> sendToDiskCache(PageRequest *): void
+<<inline>> sendToLFS(PageRequest *): void
+finish(): void
+<<virtual>> ~VFS()
PAGE_REQ
PAGE_RESP
DiskCache
#myID: int
#idInit: static int
#numDiskCache: static int
#pr_t: struct
#file_ra_state: struct
#file_t: struct
#total_pages: long
#total_usable_pages: long
#free_pages: long
#dirty_pages: long
#dirty_ratio: double
#dirty_threshold: long
#dirty_background_ratio: double
#dirty_background_threshold: long
#dirty_expire_secs: double
#bg_writeout_active: int
#disable_ra: bool
#writeout_batch: int
#diskread_batch: int
#trimCacheCounter: int
#trimCacheInt: int
#checkHealthCounter: int
#checkHealthInt: int
#writeout_list: struct pr_t *
#lru_list: struct pr_t *
#lru_list_end: struct pr_t *
#active_list: struct pr_t *
#inactive_list: struct pr_t *
#nr_scan_active: long
#nr_scan_inactive: long
#nr_active: long
#nr_inactive: long
#pages_scanned: long
#cacheAccess: PageRequest *
#bg_writeout: PageRequest *
#reqQ: list<PageRequest *> *
#prs: map<PageRequest *, pr_t *> *
#files: map<int, file_t *> *
#fileIDs[MAX_FILE]: int
#num_files: int
#cache_r_speed: double
#cache_w_speed: double
#<<virtual>> initialize(): void
#<<virtual>> handleMessage(msg:cMessage *): void
#<<inline>> handlePageReq(PageRequest *): void
#<<inline>> handleDiskAccessResp(PageRequest *): void
#<<inline>> handleCacheAccessResp(resp:PageRequest *): void
#<<inline>> handleDirtyPageExpiration(): void
#accessCache(): void
#<<inline>> readCache(PageRequest *): int
#<<inline>> writeCache(PageRequest *): void
#scheduleDiskAccess(id:long,subid:long,fid:int,
start:long,end:long,read:bool): void
#scheduleCacheAccess(id:long,subid:long,
fid:int,start:long,end:long,
read:bool): void
#walkCache_update(fid:const int,addp:pr_t *,
preop:const bool,disk:const bool): void
#<<inline>> trimAllCache(): void
#<<inline>> trimCache(file:file_t *): void
#<<inline>> preOpOnCache(fid:int,start:long,
end:long,,read:bool,
disk:bool): void
#<<inline>> getAccessSize(fid:int,start:long,
end:long,read:bool,
incache:bool): int
#<<inline>> setWriteout(fid:int,n:int): pr_t *
#<<inline>> setBGWriteoutPR(pr:pr_t *): void
#printCache(fid:int): int
#<<inline>> walkCache_countMissing(fid:int,
start:long,
end:long,
): int
#pagetable_delete(pr:pr_t *): bool
#<<inline>> lru_delete(pr:pr_t *): bool
#<<inline>> lru_push(pr:pr_t *): void
#<<inline>> lru_insert(pr1:pr_t *,pr2:pr_t *): void
#<<inline>> lru_substitute(pr1:pr_t *,pr2:pr_t *): void
#<<inline>> lru_free_pages(num:int): bool
#<<inline>> lru_free_pages(): bool
#printLRU(): int
#<<inline>> checkListHealth(): bool
#finish(): void
#~DiskCache()
LFS
ILFS
#myId: int
#degree: int
#page_size: int
#blk_size: int
#disk_size: long long
#pageReqQ: list<PageRequest *> *
#ifp: FILE *
#ofp: FILE *
#ext_t: struct
#extlist[MAX_LFILE_NUM]: ext_t *
#extentryNum[MAX_LFILE_NUM]: int
+ILFS()
-initExtLists(): void
-printExtLists(): void
#findExtEntry(fid:int,logiaddress:long): int
+<<virtual>> newReq(PageRequest *): void
+<<virtual>> dispatchNext(): BlkRequest *
+<<virtual>> finishedReq(BlkRequest *): PageRequest *
+<<virtual>> ~ILFS()
EXT3
#br_t: struct
#outstanding: int
#new_ext_size: int
#new_ext_gap: int
#brs: map<PageRequest *, br_t *> *
+EXT3(int,int,long long,int,int,const char *,
const char *,int,int)
+newReq(PageRequest *): void
+dispatchNext(): BlkRequest *
+finishedReq(BlkRequest *): PageRequest *
+~EXT3()
LFS_M
#idInit: static int
#nbDisk: static int
#lfs: ILFS *
+LFS_M()
+initialize(): void
+<<inline>> handleMessage(cMessage *): void
+<<inline>> handlePageReq(PageRequest *): void
+<<inline>> handleBlkResp(BlkRequest *): void
+<<inline>> dispatchNextDiskReq(): void
+<<inline>> sendToDisk(BlkRequest *): void
+<<inline>> sendToDiskCache(PageRequest *): void
+<<inline>> sendToVFS(PageRequest *): void
+finish(): void
+<<virtual>> ~LFS_M()
BLK_REQ
BLK_RESP
BLK_REQ
BLK_RESP
BLK_REQ
BLK_RESP
Disk
#diskSize: long int
#degree: int
#myID: int
#portno: int
#outstanding: int
#queue: IQueue *
#last_offset: long
#last_jump: long
#last_access_time: double
#seq_read_sofar: int
#return_zero_period: double
#jumpsizes[JUMPSIZECOUNT]: long
#blksizes[BLKSIZECOUNT]: long
#jumptime[2][BLKSIZECOUNT][JUMPSIZECOUNT]: double
#seqtime[2][BLKSIZECOUNT]: double
#ra_size: long
#return_zero: int
#idInit: static int
#numDisk: static int
#debugfp: FILE *
+Disk()
+getID(): int
+initialize(): void
+handleMessage(cMessage *): void
+<<inline>> handleBlockReq(BlkRequest *): void
+<<inline>> handleBlockResp(BlkRequest *): void
+dispatchJobs(): void
+checkReadahead(BlkRequest *): bool
+sendSafe(BlkRequest *): void
+readParmFile(const char *): int
+finish(): void
+<<virtual>> ~Disk()

Figure 3: Class Diagram PFSSim Package Data Server

Class Diagram MetaData Server Package

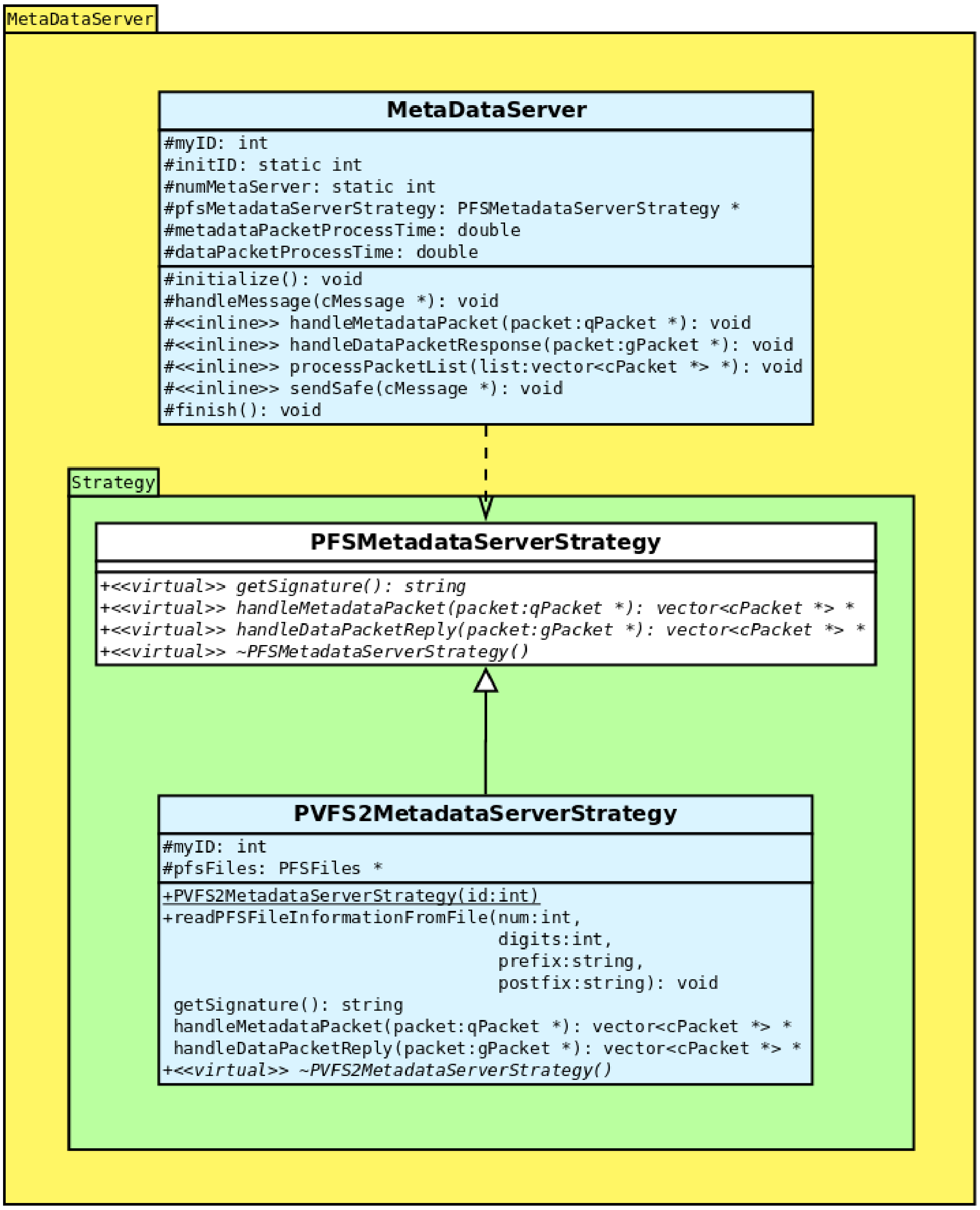
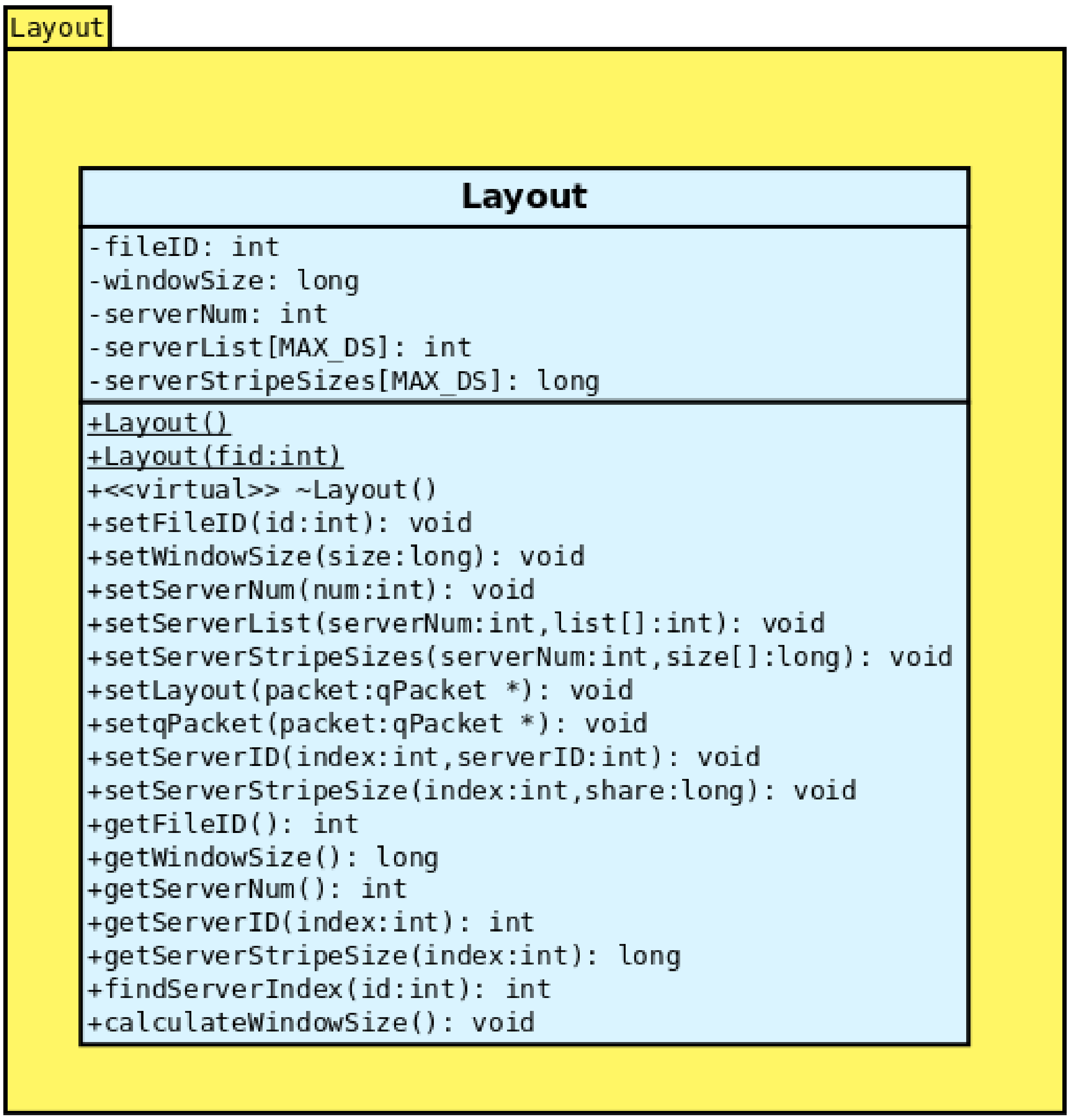

MetaDataServer
MetaDataServer
#myID: int
#initID: static int
#numMetaServer: static int
#pfsMetadataServerStrategy: PFSMetadataServerStrategy *
#metadataPacketProcessTime: double
#dataPacketProcessTime: double
#initialize(): void
#handleMessage(cMessage *): void
#<<inline>> handleMetadataPacket(packet:qPacket *): void
#<<inline>> handleDataPacketResponse(packet:gPacket *): void
#<<inline>> processPacketList(list:vector<cPacket *> *): void
#<<inline>> sendSafe(cMessage *): void
#finish(): void
Strategy
PFSMetadataServerStrategy
+<<virtual>> getSignature(): string
+<<virtual>> handleMetadataPacket(packet:qPacket *): vector<cPacket *> *
+<<virtual>> handleDataPacketReply(packet:gPacket *): vector<cPacket *> *
+<<virtual>> ~PFSMetadataServerStrategy()
PVFS2MetadataServerStrategy
#myID: int
#pfsFiles: PFSFiles *
+PVFS2MetadataServerStrategy(id:int)
+readPFSFileInformationFromFile(num:int, digits:int, prefix:string, postfix:string): void
getSignature(): string
handleMetadataPacket(packet:qPacket *): vector<cPacket *> *
handleDataPacketReply(packet:gPacket *): vector<cPacket *> *
+<<virtual>> ~PVFS2MetadataServerStrategy()

Figure 4: Class Diagram PFSSim Package MetaData Server

Class Diagram Layout Package



```
-fileID: int
-windowSize: long
-serverNum: int
-serverList[MAX_DS]: int
-serverStripeSizes[MAX_DS]: long
```

```
+Layout()
+Layout(fid:int)
+<<virtual>> ~Layout()
+setFileID(id:int): void
+setWindowSize(size:long): void
+setServerNum(num:int): void
+setServerList(serverNum:int,list[]:int): void
+setServerStripeSizes(serverNum:int,size[]:long): void
+setLayout(packet:qPacket *): void
+setqPacket(packet:qPacket *): void
+setServerID(index:int,serverID:int): void
+setServerStripeSize(index:int,share:long): void
+getFileID(): int
+getWindowSize(): long
+getServerNum(): int
+getServerID(index:int): int
+getServerStripeSize(index:int): long
+findServerIndex(id:int): int
+calculateWindowSize(): void
```

Figure 5: Class Diagram PFSSim Package Layout

Class Diagram Proxy Package

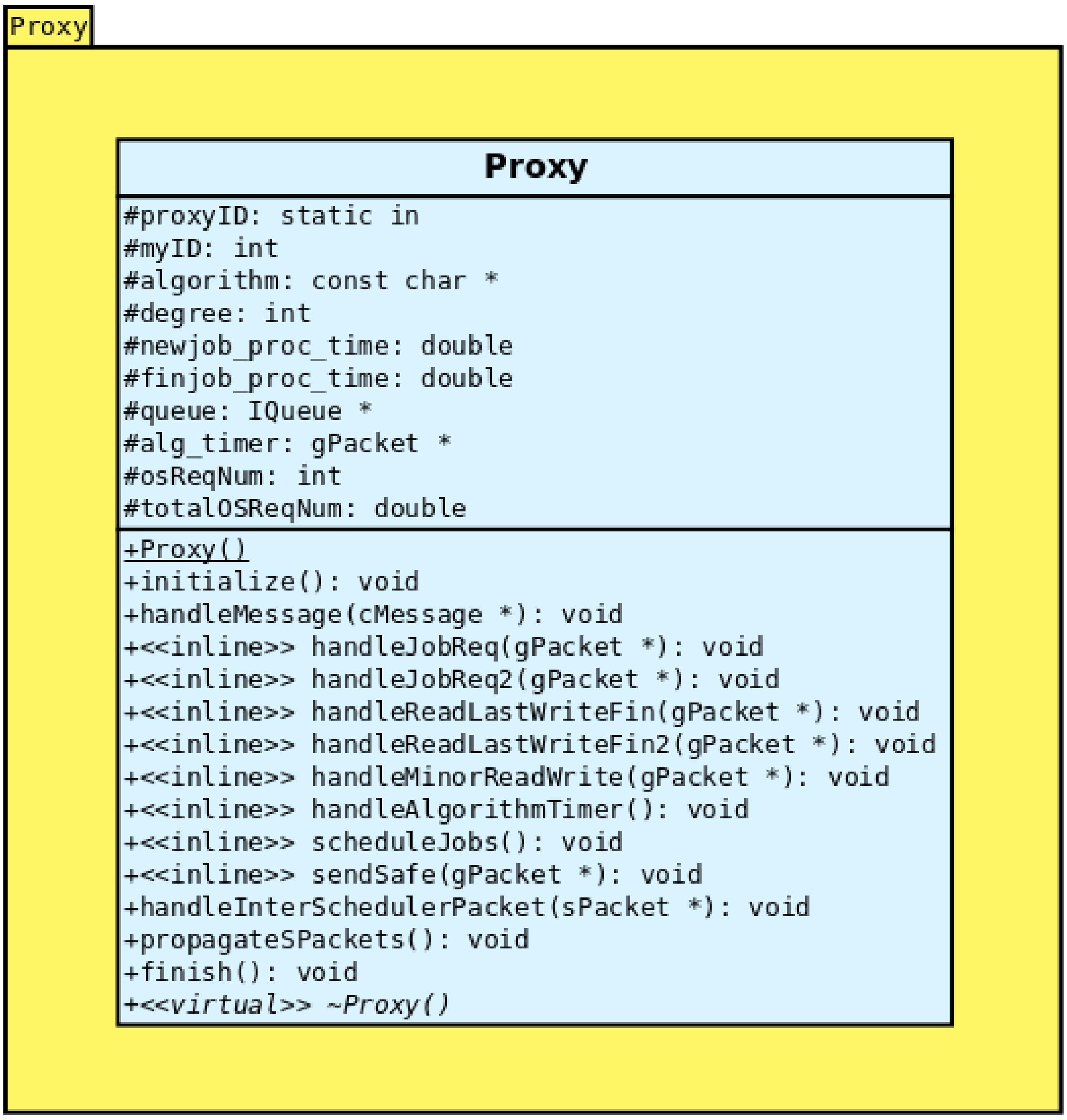

Proxy
Proxy
#proxyID: static in
#myID: int
#algorithm: const char *
#degree: int
#newjob_proc_time: double
#finjob_proc_time: double
#queue: IQueue *
#alg_timer: gPacket *
#osReqNum: int
#totalOSReqNum: double
+Proxy()
+initialize(): void
+handleMessage(cMessage *): void
+<<inline>> handleJobReq(gPacket *): void
+<<inline>> handleJobReq2(gPacket *): void
+<<inline>> handleReadLastWriteFin(gPacket *): void
+<<inline>> handleReadLastWriteFin2(gPacket *): void
+<<inline>> handleMinorReadWrite(gPacket *): void
+<<inline>> handleAlgorithmTimer(): void
+<<inline>> scheduleJobs(): void
+<<inline>> sendSafe(gPacket *): void
+handleInterSchedulerPacket(sPacket *): void
+propagateSPackets(): void
+finish(): void
+<<virtual>> ~Proxy()

Figure 6: Class Diagram PFSSim Package Proxy

Class Diagram Scheduler Package

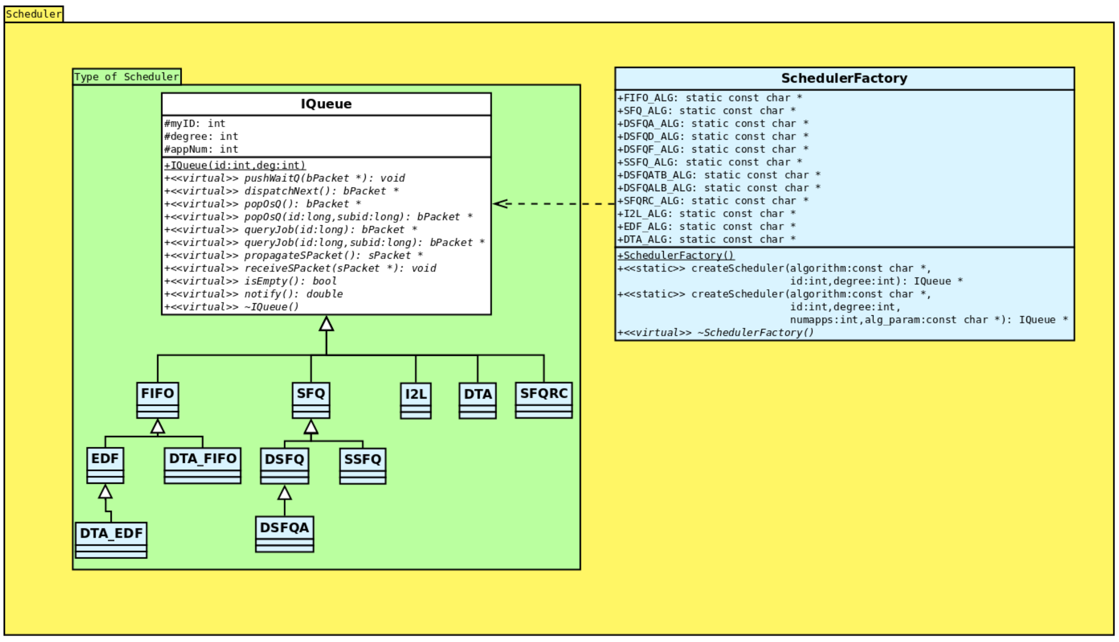

Scheduler
Type of Scheduler
IQueue
#myID: int
#degree: int
#appNum: int
+IQueue(id:int,deg:int)
+<<virtual>> pushWaitQ(bPacket *): void
+<<virtual>> dispatchNext(): bPacket *
+<<virtual>> popOsQ(): bPacket *
+<<virtual>> popOsQ(id:long,subid:long): bPacket *
+<<virtual>> queryJob(id:long): bPacket *
+<<virtual>> queryJob(id:long,subid:long): bPacket *
+<<virtual>> propagateSPacket(): sPacket *
+<<virtual>> receiveSPacket(sPacket *): void
+<<virtual>> isEmpty(): bool
+<<virtual>> notify(): double
+<<virtual>> ~IQueue()
FIFO
SFQ
I2L
DTA
SFQRC
EDF
DTA_FIFO
DSFQ
SSFQ
DTA_EDF
DSFQA
SchedulerFactory
+FIFO_ALG: static const char *
+SFQ_ALG: static const char *
+DSFQA_ALG: static const char *
+DSFQD_ALG: static const char *
+DSFQF_ALG: static const char *
+SSFQ_ALG: static const char *
+DSFQATB_ALG: static const char *
+DSFQALB_ALG: static const char *
+SFQRC_ALG: static const char *
+I2L_ALG: static const char *
+EDF_ALG: static const char *
+DTA_ALG: static const char *
+SchedulerFactory()
+<<static>> createScheduler(algorithm:const char *, id:int,degree:int): IQueue *
+<<static>> createScheduler(algorithm:const char *, id:int,degree:int, numapps:int,alg_param:const char *): IQueue *
+<<virtual>> ~SchedulerFactory()

Figure 7: Class Diagram PFSSim Package Schedulers