```
**********************************************************
    1275 Mon Feb  3 15:34:52 2014
new/usr/src/cmd/stat/Makefile
arcstat
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# cmd/stat/Makefile
#

include ../Makefile.cmd

SUBDIRS=        arcstat iostat mpstat vmstat fsstat kstat
SUBDIRS=        iostat mpstat vmstat fsstat kstat

all :=          TARGET = all
install :=      TARGET = install
clean :=        TARGET = clean
clobber :=      TARGET = clobber
lint :=         TARGET = lint
_msg :=         TARGET = _msg

.KEEP_STATE:

all install lint clean clobber _msg: $(SUBDIRS)

$(SUBDIRS): FRC
        @cd $@; pwd; $(MAKE) $(MFLAGS) $(TARGET)

FRC:
```

```
**********************************************************
    1698 Mon Feb  3 15:34:52 2014
new/usr/src/cmd/stat/arcstat/Makefile
arcstat
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

PROG = arcstat
OBJS = arcstat.o
SRCS =$(OBJS:%.o=%.c) $(COMMON_SRCS)

include $(SRC)/cmd/Makefile.cmd
include $(SRC)/cmd/stat/Makefile.stat

LDLIBS += -lavl -lcmdutils -ldevinfo -lgen -lkstat
CFLAGS += $(CCVERBOSE) -I${STATCOMMONDIR}
CERRWARN += -_gcc=-Wno-uninitialized
CERRWARN += -_gcc=-Wno-switch

CPPFLAGS_sparc += -I$(SRC)/uts/sfmmu
CPPFLAGS_sparc += -I$(SRC)/uts/sun4u/sunfire
CPPFLAGS += $(CPPFLAGS_$(MACH))

FILEMODE= 0555

lint := LINTFLAGS = -muxs -I$(STATCOMMONDIR)

.KEEP_STATE:

all: $(PROG)

install: all $(ROOTPROG)

$(PROG): $(OBJS) $(COMMON_OBJS)
        $(LINK.c) -o $(PROG) $(OBJS) $(COMMON_OBJS) $(LDLIBS)
        $(POST_PROCESS)

%.o : $(STATCOMMONDIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

clean:
        -$(RM) $(OBJS) $(COMMON_OBJS)
```



```
lint: lint_SRCS

include $(SRC)/cmd/Makefile.targ
#endif /* ! codereview */
```

```
**********************************************************
   20800 Mon Feb  3 15:34:53 2014
new/usr/src/cmd/stat/arcstat/arcstat.c
arcstat
**********************************************************
```

```
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2014 David Hoeppner.  All rights reserved.
 */

/*
 * Display ZFS ARC statistics.
 *
 * Based on work by Neelakanth Nadgir and Mike Harsch.
 */
#include <sys/list.h>
#include <assert.h>
#include <errno.h>
#include <kstat.h>
#include <libintl.h>
#include <limits.h>
#include <locale.h>
#include <poll.h>
#include <stddef.h>
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>

#include "statcommon.h"

char *cmdname = "arcstat";      /* Name of this command */
int caught_cont = 0;            /* Have caught a SIGCONT */

/* Saved command line options */
static boolean_t g_fflg = B_FALSE;      /* custom header fields */
static boolean_t g_oflg = B_FALSE;      /* write output file */
static boolean_t g_rflg = B_FALSE;      /* raw output */
static boolean_t g_vflg = B_FALSE;      /* verbose help */
static boolean_t g_xflg = B_FALSE;      /* extended header */

/* Time in seconds between snapshots */
int             interval = 1;

static list_t   fields_list;
static char     *separator = "  ";
static char     *output_file = NULL;
static char     *hdr;

typedef struct _arcstat_delta {
        int64_t ad_hits;
        int64_t ad_miss;
        int64_t ad_read;
        int64_t ad_hit_percent;
        int64_t ad_miss_percent;
        int64_t ad_dhit;
        int64_t ad_dmis;
```



```
        int64_t ad_dread;
        int64_t ad_dh_percent;
        int64_t ad_dm_percent;
        int64_t ad_phit;
        int64_t ad_pmis;
        int64_t ad_pread;
        int64_t ad_ph_percent;
        int64_t ad_pm_percent;
        int64_t ad_mhit;
        int64_t ad_mmis;
        int64_t ad_mread;
        int64_t ad_mh_percent;
        int64_t ad_mm_percent;
        int64_t ad_arcsz;
        int64_t ad_c;
        int64_t ad_mfu;
        int64_t ad_mru;
        int64_t ad_mfug;
        int64_t ad_mrug;
        int64_t ad_eskip;
        int64_t ad_rmiss;
        int64_t ad_mtxmis;
} arcstat_delta_t;

typedef struct arcstat_snapshot {
        kstat_t as_arcstats;
        uint64_t as_hits;
        uint64_t as_misses;
        uint64_t as_demand_data_hits;
        uint64_t as_demand_data_misses;
        uint64_t as_demand_metadata_hits;
        uint64_t as_demand_metadata_misses;
        uint64_t as_prefetch_data_hits;
        uint64_t as_prefetch_data_misses;
        uint64_t as_prefetch_metadata_hits;
        uint64_t as_prefetch_metadata_misses;
        uint64_t as_size;
        uint64_t as_c;
        uint64_t as_mfu_hits;
        uint64_t as_mru_hits;
        uint64_t as_mru_ghost_hits;
        uint64_t as_mfu_ghost_hits;
        uint64_t as_evict_skip;
        uint64_t as_recycle_miss;
        uint64_t as_mutex_miss;
        uint64_t as_l2_hits;
        uint64_t as_l2_misses;
        uint64_t as_l2_size;
        uint64_t as_l2_read_bytes;
} arcstat_snapshot_t;

typedef struct arcstat_hdr_field {
        list_node_t     ahf_next;
        char            *ahf_name;
        struct arcstat_field *ahf_desc;
} arcstat_hdr_field_t;

typedef enum field_type {
        as_time,
        as_hits,
        as_miss,
        as_read,
        as_hit_percent,
        as_miss_percent,
        as_dhit,
        as_dmis,
```

```
128         as_dread,
129         as_dh_percent,
130         as_dm_percent,
131         as_phit,
132         as_pmis,
133         as_pread,
134         as_ph_percent,
135         as_pm_percent,
136         as_mhit,
137         as_mmis,
138         as_mread,
139         as_mh_percent,
140         as_mm_percent,
141         as_arcsz,
142         as_c,
143 } field_type_t;

145 /* XXX */
146 struct arcstat_field {
147         char    *af_name;
148         char    *af_description;
149         int     af_length;
150         int     af_interval;
151         field_type_t    af_type;
152 } arcstat_fields[] = {
153         { "time", "Time", 8, -1, as_time },
154         { "hits", "ARC reads per second", 4, 1000, as_hits },
155         { "miss", "ARC misses per second", 4, 1000, as_miss },
156         { "read", "Total ARC accesses per second", 4, 1000, as_read },
157         { "hit%", "ARC Hit percentage", 4, 100, as_hit_percent },
158         { "miss%", "ARC miss percentage", 5, 100, as_miss_percent },
159         { "dhit", "Demand Data hits per second", 4, 1000, as_dhit },
160         { "dmis", "Demand Data misses per second", 4, 1000, as_dmis },
161         { "dread", "Demand data accesses per second", 5, 1000, as_dread },
162         { "dh%", "Demand Data hit percentage", 3, 100, as_dh_percent  },
163         { "dm%", "Demand Data miss percentage", 3, 100, as_dm_percent },
164         { "phit", "Prefetch hits per second", 4, 1000, as_phit },
165         { "pmis", "Prefetch misses per second", 4, 1000, as_pmis },
166         { "ph%", "Prefetch hits percentage", 3, 100, as_ph_percent },
167         { "pm%", "Prefetch miss percentage", 3, 100, as_pm_percent },
168         { "mhit", "Metadata hits per second", 4, 1000, as_mhit },
169         { "mmis", "Metadata misses per second", 4, 1000, as_mmis },
170         { "mread", "Metadata accesses per second", 4, 1000, as_mread },
171         { "mh%", "Metadata hit percentage", 3, 100, as_mh_percent },
172         { "mm%", "Metadata miss percentage", 3, 100, as_mm_percent },
173         { "arcsz", "ARC Size", 5, 1024, as_arcsz },
174         { "c", "ARC Target Size", 4, 1024, as_c },
175         { "mfu", "MFU List hits per second", 4, 1000 },
176         { "mru", "MRU List hits per second", 4, 1000 },
177         { "mfug", "MFU Ghost List hits per second", 4, 1000 },
178         { "mrug", "MRU Ghost List hits per second", 4, 1000 },
179         { "eskip", "evict_skip per second", 5, 1000 },
180         { "mtxmis", "mutex_miss per second", 6, 1000 },
181         { "rmis", "recycle_miss per second", 4, 1000 },
182         { "pread", "Prefetch accesses per second", 5, 1000 },
183         { "l2hits", "L2ARC hits per second", 6, 1000 },
184         { "l2miss", "L2ARC misses per second", 6, 1000 },
185         { "l2read", "Total L2ARC accesses per second", 6, 1000 },
186         { "l2hit%", "L2ARC access hit percentage", 6, 100 },
187         { "l2miss%", "L2ARC access miss percentage", 7, 100 },
188         { "l2size", "Size of the L2ARC", 6, 1024 },
189         { "l2bytes", "bytes read per second from the L2ARC", 7, 1024 },
190         { NULL, NULL, -1, -1 },
191 };

193 /*
```



```
194  * Print usage.
195  */
196 static void
197 usage(void)
198 {
199         (void) fprintf(stderr, gettext(
200             "Usage: arcstat [-hvxr] [-f fields] [-o file] [-s string] "
201             "[interval [count]]\n\n"));

203         if (g_vflg) {
204                 int     i = 0;

206                 (void) fprintf(stderr, gettext(
207                     "Field definitions are as follows:\n"));

209                 for (;arcstat_fields[i].af_name != NULL; i++) {
210                         (void) fprintf(stderr, "%11s : ",
211                             arcstat_fields[i].af_name);
212                         (void) fprintf(stderr, "%s\n", gettext(
213                             arcstat_fields[i].af_description));
214                 }
215         }
216 }

218 /*
219  * Print header.
220  */
221 static void
222 printhdr(int sig)
223 {
224         arcstat_hdr_field_t     *hdr_field;

226         /*
227          * Reenable the signal.
228          */
229         if (sig)
230                 (void) signal(SIGCONT, printhdr);
231         if (sig == SIGCONT)
232                 caught_cont = 1;

234         /*
235          * Print fields in fields list for this header.
236          */
237         hdr_field = list_head(&fields_list);
238         while (hdr_field != NULL) {
239                 if (g_rflg) {
240                         printf("%s%s", hdr_field->ahf_name, separator);
241                 } else {
242                         printf("%*s%s", hdr_field->ahf_desc->af_length,
243                             hdr_field->ahf_name, separator);
244                 }

246                 hdr_field = list_next(&fields_list, hdr_field);
247         }

249         (void) putchar('\n');
250 }

252 /*
253  * Pretty print values.
254  */
255 static void
256 printvals(arcstat_delta_t *delta)
257 {
258         arcstat_hdr_field_t     *hdr_field;
```

```
#define ARCSTAT_PRINT_VALUE(field, delta, value)        \
        (void) printf("%*llu%s", field->ahf_desc->af_length, delta->ad_##value,

        /*
         * Print values for fields.
         */
        hdr_field = list_head(&fields_list);
        while (hdr_field != NULL) {
                switch (hdr_field->ahf_desc->af_type) {
                case as_time: {
                        time_t  t = time(NULL);
                        char    dstr[64];
                        char    *fmt = "%H:%M:%S";
                        int     len;

                        len = strftime(dstr, sizeof (dstr), fmt, localtime(&t));
                        if (len > 0)
                                (void) printf("%s%s", dstr, separator);
                        break;
                };
                case as_miss:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, miss);
                        break;
                case as_read:
                        (void) printf("%*llu%s", hdr_field->ahf_desc->af_length,
                        break;
                case as_hit_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, hit_percent);
                        break;
                case as_miss_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, miss_percent);
                        break;
                case as_dhit:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, dhit);
                        break;
                case as_dmis:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, dmis);
                        break;
                case as_dread:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, dread);
                        break;
                case as_dh_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, dh_percent);
                        break;
                case as_dm_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, dm_percent);
                        break;
                case as_phit:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, phit);
                        break;
                case as_pmis:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, pmis);
                        break;
                case as_pread:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, pread);
                        break;
                case as_ph_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, ph_percent);
                        break;
                case as_pm_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, pm_percent);
                        break;
                case as_mhit:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, mhit);
                        break;
                case as_mmis:
```



```
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, mmis);
                        break;
                case as_mread:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, mread);
                        break;
                case as_mh_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, mh_percent);
                        break;
                case as_mm_percent:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, mm_percent);
                        break;
                case as_arcsz:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, arcsz);
                        break;
                case as_c:
                        ARCSTAT_PRINT_VALUE(hdr_field, delta, c);
                        break;
                default:
                        break;
                }

                // number_to_scaled_string

                hdr_field = list_next(&fields_list, hdr_field);
        }

        (void) putchar('\n');
}

arcstat_snapshot_t *
arcstat_acquire_snapshot(kstat_ctl_t *kc)
{
        arcstat_snapshot_t      *snapshot;
        kstat_named_t           *knp;
        kstat_t                 *ksp;

        if ((ksp = kstat_lookup(kc, "zfs", 0, "arcstats")) == NULL)
                return (NULL);

        if (kstat_read(kc, ksp, NULL) == -1)
                return (NULL);

        snapshot = safe_alloc(sizeof (arcstat_snapshot_t));

/* XXX snapshot */
#define ARC_KSTAT_DATA_LOOKUP(name)                                     \
        knp = (kstat_named_t *)kstat_data_lookup(ksp, #name);           \
        if (knp == NULL)                                                \
                return (NULL);                                          \
        snapshot->as_##name = knp->value.ui64;

        ARC_KSTAT_DATA_LOOKUP(hits);
        ARC_KSTAT_DATA_LOOKUP(misses);
        ARC_KSTAT_DATA_LOOKUP(demand_data_hits);
        ARC_KSTAT_DATA_LOOKUP(demand_data_misses);
        ARC_KSTAT_DATA_LOOKUP(demand_metadata_hits);
        ARC_KSTAT_DATA_LOOKUP(demand_metadata_misses);
        ARC_KSTAT_DATA_LOOKUP(prefetch_data_hits);
        ARC_KSTAT_DATA_LOOKUP(prefetch_data_misses);
        ARC_KSTAT_DATA_LOOKUP(prefetch_metadata_hits);
        ARC_KSTAT_DATA_LOOKUP(prefetch_metadata_misses);
        ARC_KSTAT_DATA_LOOKUP(size);
        ARC_KSTAT_DATA_LOOKUP(c);
        ARC_KSTAT_DATA_LOOKUP(mfu_hits);
        ARC_KSTAT_DATA_LOOKUP(mru_hits);
        ARC_KSTAT_DATA_LOOKUP(mru_ghost_hits);
```

```
392         ARC_KSTAT_DATA_LOOKUP(mfu_ghost_hits);
393         ARC_KSTAT_DATA_LOOKUP(evict_skip);
394         ARC_KSTAT_DATA_LOOKUP(recycle_miss);
395         ARC_KSTAT_DATA_LOOKUP(mutex_miss);
396         ARC_KSTAT_DATA_LOOKUP(l2_hits);
397         ARC_KSTAT_DATA_LOOKUP(l2_misses);
398         ARC_KSTAT_DATA_LOOKUP(l2_size);
399         ARC_KSTAT_DATA_LOOKUP(l2_read_bytes);

401         return (snapshot);
402 }

404 static void
405 arcstat_free_snapshot(arcstat_snapshot_t *old)
406 {
407 }

409 static void
410 arcstat_chain_update(kstat_ctl_t *kc)
411 {
412         int     ret;

414         ret = kstat_chain_update(kc);
415         if (ret != 0) {
416                 (void) printf("<State Changed>\n");
417         }
418 }

420 static void
421 free_field_list(list_t list)
422 {
423         arcstat_hdr_field_t     *hdr_field;

425         hdr_field = list_head(&list);
426         while (hdr_field != NULL) {

428                 hdr_field = list_next(&list, hdr_field);
429         }
430 }

432 static arcstat_delta_t *
433 arcstat_calculate_delta(arcstat_snapshot_t *old, arcstat_snapshot_t *new)
434 {
435         arcstat_delta_t *delta;

437         assert(old != NULL && new != NULL);

439         delta = safe_alloc(sizeof (arcstat_delta_t));

441         (void) memset(delta, 0, sizeof (arcstat_delta_t));

443 #define ARC_SNAPSHOT_DIFF(O, N, S)                      \
444         (N->as_##S - O->as_##S)

446         delta->ad_hits = ARC_SNAPSHOT_DIFF(old, new, hits) / interval;
447         delta->ad_miss = ARC_SNAPSHOT_DIFF(old, new, misses) / interval;

449         delta->ad_read = delta->ad_hits + delta->ad_miss;
450         if (delta->ad_read > 0) {
451                 delta->ad_hit_percent = 100 * (delta->ad_hits / delta->ad_read);
452                 delta->ad_miss_percent = 100 - delta->ad_hit_percent;
453         }

455         delta->ad_dhit = (ARC_SNAPSHOT_DIFF(old, new, demand_data_hits) +
456             ARC_SNAPSHOT_DIFF(old, new, demand_metadata_hits)) / interval;
457         delta->ad_dmis = (ARC_SNAPSHOT_DIFF(old, new, demand_data_misses) +
```



```
458             ARC_SNAPSHOT_DIFF(old, new, demand_metadata_misses)) / interval;

460         delta->ad_dread = delta->ad_dhit + delta->ad_dmis;
461         if (delta->ad_dread > 0) {
462                 delta->ad_dh_percent = 100 * (delta->ad_dhit / delta->ad_dread);
463                 delta->ad_dm_percent = 100 - delta->ad_dh_percent;
464         }

466         delta->ad_phit = (ARC_SNAPSHOT_DIFF(old, new, prefetch_data_hits) +
467             ARC_SNAPSHOT_DIFF(old, new, prefetch_metadata_hits)) / interval;
468         delta->ad_pmis = (ARC_SNAPSHOT_DIFF(old, new, prefetch_data_misses) +
469             ARC_SNAPSHOT_DIFF(old, new, prefetch_metadata_misses)) / interval;

471         delta->ad_pread = delta->ad_phit + delta->ad_pmis;
472         if (delta->ad_pread > 0) {
473                 delta->ad_ph_percent = 100 * (delta->ad_phit / delta->ad_pread);
474                 delta->ad_pm_percent = 100 - delta->ad_ph_percent;
475         }

477         delta->ad_mhit = (ARC_SNAPSHOT_DIFF(old, new, prefetch_metadata_hits) +
478             ARC_SNAPSHOT_DIFF(old, new, demand_metadata_hits)) / interval;
479         delta->ad_mmis = (ARC_SNAPSHOT_DIFF(old, new,
480             prefetch_metadata_misses) + ARC_SNAPSHOT_DIFF(old, new,
481             demand_metadata_misses)) / interval;

483         delta->ad_mread = delta->ad_mhit + delta->ad_mmis;
484         if (delta->ad_mread > 0) {
485                 delta->ad_mh_percent = 100 * (delta->ad_mhit / delta->ad_mread);
486                 delta->ad_mm_percent = 100 - delta->ad_mh_percent;
487         }

489         delta->ad_arcsz = new->as_size;
490         delta->ad_c = new->as_c;
491         delta->ad_mfu = ARC_SNAPSHOT_DIFF(old, new, mfu_hits) / interval;
492         delta->ad_mru = ARC_SNAPSHOT_DIFF(old, new, mru_hits) / interval;
493         delta->ad_mfug = ARC_SNAPSHOT_DIFF(old, new, mfu_ghost_hits) /
494             interval;
495         delta->ad_mrug = ARC_SNAPSHOT_DIFF(old, new, mru_ghost_hits) /
496             interval;
497         delta->ad_eskip = ARC_SNAPSHOT_DIFF(old, new, evict_skip) / interval;
498         delta->ad_rmiss = ARC_SNAPSHOT_DIFF(old, new, recycle_miss) / interval;
499         delta->ad_mtxmis = ARC_SNAPSHOT_DIFF(old, new, mutex_miss) / interval;

501         /* XXX L2 cache */

503         return (delta);
504 }

506 int
507 main(int argc, char **argv)
508 {
509         int             c;
510         int             i = 0;
511         int             iter = 1;
512         char            *endptr;
513         int             infinite_cycles = 0;
514         kstat_ctl_t     *kc;
515         hrtime_t        start_n;
516         hrtime_t        period_n;
517         list_t          invalid_list;
518         list_t          incompt_list;
519         boolean_t       l2exist = B_FALSE;
520         arcstat_hdr_field_t     *hdr_field;
521         arcstat_snapshot_t      *old = NULL;
522         arcstat_snapshot_t      *new = NULL;
```

```
525         (void) setlocale(LC_ALL, "");
526 #if !defined(TEXT_DOMAIN)       /* Should be defined by cc -D */
527 #define TEXT_DOMAIN "SYS_TEST"  /* Use this only if it wasn't */
528 #endif
529         (void) textdomain(TEXT_DOMAIN);

531         /*
532          * Parse command line arguments.
533          */
534         while ((c = getopt(argc, argv, "h?f:o:rs:vx")) != EOF) {
535                 switch (c) {
536                 case 'h':
537                 case '?':
538                         usage();
539                         exit(0);
540                         break;
541                 case 'f':
542                         g_fflg = B_TRUE;
543                         hdr = safe_strdup(optarg);
544                         break;
545                 case 'o':
546                         g_oflg = B_TRUE;
547                         output_file = (char *)optarg;
548                 case 'r':
549                         g_rflg = B_TRUE;
550                         break;
551                 case 's':
552                         separator = (char *)optarg;
553                         break;
554                 case 'v':
555                         g_vflg = B_TRUE;
556                         usage();
557                         exit(0);
558                         break;
559                 case 'x':
560                         g_xflg = B_TRUE;
561                         break;
562                 default:
563                         break;
564                 }
565         }

567         /*
568          * Select the standard, extended or user supplied header fields.
569          */
570         if (!g_fflg) {
571                 if (g_xflg) {
572                         hdr = safe_strdup("time,mfu,mru,mfug,mrug,eskip,mtxmis,r
573                 } else {
574                         hdr = safe_strdup("time,read,miss,miss%,dmis,dm%,pmis,pm
575                 }
576         }

578         /*
579          * Interval and count.
580          */
581         if (argc > optind) {
582                 interval = (int)strtol(argv[optind], &endptr, 10);
583                 if (*endptr != NULL)
584                         usage();
585                 period_n = (hrtime_t)interval * NANOSEC;
586                 if (argc > optind + 1) {
587                         iter = (unsigned int)strtoul
588                             (argv[optind + 1], &endptr, 10);
589                         if (*endptr != NULL || iter < 0)
```



```
590                                 usage();
591                         if (iter == 0)
592                                 return (0);     /* XXX */
593                 } else {
594                         infinite_cycles = 1;
595                 }
596         }

598         /*
599          * Need to know if there is a L2 cache.
600          */
601         kc = open_kstat();

603         /* XXX L2 cache */

605         /* Valid field names */
606         list_create(&fields_list, sizeof (arcstat_hdr_field_t),
607             offsetof(arcstat_hdr_field_t, ahf_next));

609         /* Invalid field names */
610         list_create(&invalid_list, sizeof (arcstat_hdr_field_t),
611             offsetof(arcstat_hdr_field_t, ahf_next));

613         /* Invalid field names if no L2 ARC */
614         list_create(&incompt_list, sizeof (arcstat_hdr_field_t),
615             offsetof(arcstat_hdr_field_t, ahf_next));

617         while ((endptr = (char *)strsep(&hdr, ",")) != NULL) {
618                 boolean_t       found_field = B_FALSE;
619                 int             i = 0;

621                 for (; arcstat_fields[i].af_name != NULL; i++) {
622                         if (strcmp(arcstat_fields[i].af_name, endptr) == 0) {
623                                 found_field = B_TRUE;
624                                 break;
625                         }
626                 }

628                 /*
629                  * Allocate a new header field and link it to the fields table.
630                  */
631                 hdr_field = safe_alloc(sizeof (arcstat_hdr_field_t));
632                 hdr_field->ahf_name = safe_strdup(endptr);
633                 hdr_field->ahf_desc = &arcstat_fields[i];

635                 list_link_init(&hdr_field->ahf_next);

637                 /*
638                  * Add valid fields to fields_list or to the list of invalid fie
639                  */
640                 if (found_field) {
641                         list_insert_tail(&fields_list, hdr_field);
642                 } else {
643                         list_insert_tail(&invalid_list, hdr_field);
644                 }
645         }

647         /*
648          * User supplied an invalid field per cmdline.
649          */
650         if (!list_is_empty(&invalid_list)) {
651                 (void) fprintf(stderr, "%s -- ", gettext(
652                     "Invalid column definition!"));

654                 hdr_field = list_head(&invalid_list);
655                 while (hdr_field != NULL) {
```

```
656                         (void) fprintf(stderr, "%s ", hdr_field->ahf_name);

658                         hdr_field = list_next(&invalid_list, hdr_field);
659                 }

661                 (void) fprintf(stderr, "\n\n");

663                 //free_field_list(invalid_list);

665                 usage();
666                 exit(2);
667         }

669         /*
670          * User supplied an L2 cache field, but we have no L2 cache.
671          */
672         if (!list_is_empty(&incompt_list)) {
673                 (void) fprintf(stderr, "%s -- ", gettext(
674                     "Incompatible field specified!"));

676                 hdr_field = list_head(&incompt_list);
677                 while (hdr_field != NULL) {
678                         (void) fprintf(stderr, "%s ", hdr_field->ahf_name);

680                         hdr_field = list_next(&incompt_list, hdr_field);
681                 }

683                 (void) fprintf(stderr, "\n\n");

685                 usage();
686                 exit(2);
687         }

689         /* We should have at least one valid field */
690         assert(!list_is_empty(&fields_list));

692         /*
693          * If we need write to a file, try to open it.
694          */
695         if (g_oflg) {

697         }

699         // (void) sigset(SIGCONT, printhdr);
700         /* Set up handler for SIGCONT */
701         if (signal(SIGCONT, cont_handler) == SIG_ERR)
702                 fail(1, "signal failed");

704         start_n = gethrtime();

706         new = arcstat_acquire_snapshot(kc);
707         while (infinite_cycles || iter > 0) {
708                 arcstat_delta_t *delta;

710                 arcstat_free_snapshot(old);
711                 old = new;
712                 new = arcstat_acquire_snapshot(kc);

714                 arcstat_chain_update(kc);

716                 /* XXX */
717                 if (i % 20 == 0)
718                         printhdr(0);
719                 i++;

721                 delta = arcstat_calculate_delta(old, new);
```



```
722                 printvals(delta);

724                 if (!infinite_cycles && --iter < 1)
725                         break;

727                 sleep_until(&start_n, period_n, infinite_cycles, &caught_cont);
728         }

730         (void) kstat_close(kc);

732         arcstat_free_snapshot(old);
733         arcstat_free_snapshot(new);

735         free(hdr);

737         return (0);
738 }

740 #define NUMBER_WIDTH    64
741 typedef char numbuf_t[NUMBER_WIDTH];

743 /* Copied from du.c */
744 static char *
745 number_to_scaled_string(
746         numbuf_t buf,                   /* put the result here */
747         unsigned long long number,      /* convert this number */
748         unsigned long long unit_from,   /* number of byes per input unit */
749         unsigned long long scale)       /* 1024 (-h) or 1000 (-H) */
750 {
751         unsigned long long save = 0;
752         char *M = "KMGTPE"; /* Measurement: kilo, mega, giga, tera, peta, exa */
753         char *uom = M;  /* unit of measurement, initially 'K' (=M[0]) */

755         if ((long long)number == (long long) -1) {
756                 (void) strcpy(buf, "-1");
757                 return (buf);
758         }

760         /*
761          * Convert number from unit_from to given scale (1024 or 1000)
762          * This means multiply number with unit_from and divide by scale.
763          * if number is large enough, we first divide and then multiply
764          * to avoid an overflow (large enough here means 100 (rather arbitrary
765          * value) times scale in order to reduce rounding errors)
766          * otherwise, we first multiply and then divide to avoid an underflow.
767          */
768         if (number >= 100L * scale) {
769                 number = number / scale;
770                 number = number * unit_from;
771         } else {
772                 number = number * unit_from;
773                 number = number / scale;
774         }

776         /*
777          * Now we have number as a count of scale units.
778          * Stop scaling when we reached exa bytes, then something is
779          * probably wrong with our number.
780          */
781         while ((number >= scale) && (*uom != 'E')) {
782                 uom++;  /* Next unit of measurement */
783                 save = number;
784                 number = (number + (scale / 2)) / scale;
785         }

787         /* Check if we should output a decimal place after the point */
```

```
788         if (save && ((save / scale) < 10)) {
789                 /* sprintf() will round for us */
790                 float fnum = (float)save / scale;
791                 (void) sprintf(buf, "%.1f%c", fnum, *uom);
792         } else {
793                 (void) sprintf(buf, "%llu%c", number, *uom);
794         }
795         return (buf);
796 }
797 #endif /* ! codereview */
```

```
**********************************************************
   11816 Mon Feb  3 15:34:53 2014
new/usr/src/cmd/stat/common/acquire.c
arcstat
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    3376 Mon Feb  3 15:34:53 2014
new/usr/src/cmd/stat/common/common.c
arcstat
**********************************************************
______unchanged_portion_omitted_

  54 /*
  55  * Sleep until *wakeup + interval, keeping cadence where desired
  56  *
  57  * *wakeup -    The time we last wanted to wake up. Updated.
  58  * interval -   We want to sleep until *wakeup + interval
  59  * forever -    Running for infinite periods, so cadence not important
  60  * *caught_cont - Global set by signal handler if we got a SIGCONT
  61  */
  62 void
  63 sleep_until(hrtime_t *wakeup, hrtime_t interval, int forever,
  64     int *caught_cont)
  65 {
  66         hrtime_t now, pause, pause_left;
  67         struct timespec pause_tv;
  68         int status;

  70         now = gethrtime();
  71         pause = *wakeup + interval - now;

  73         if (pause <= 0 || pause < (interval / 4))
  74                 if (forever || *caught_cont) {
  75                         /* Reset our cadence (see comment below) */
  76                         *wakeup = now + interval;
  77                         pause = interval;
  78                 } else {
  79                         /*
  80                          * If we got here, then the time between the
  81                          * output we just did, and the scheduled time
  82                          * for the next output is < 1/4 of our requested
  83                          * interval AND the number of intervals has been
  84                          * requested AND we have never caught a SIGCONT
  85                          * (so we have never been suspended).  In this
  86                          * case, we'll try to stay to the desired
  87                          * cadence, and we will pause for 1/2 the normal
  88                          * interval this time.
  89                          */
  90                         pause = interval / 2;
  91                         *wakeup += interval;
  92                 }
  93         else
  94                 *wakeup += interval;
  95         if (pause < 1000)
  96                 /* Near enough */
  97                 return;

  99         /* Now do the actual sleep */
 100         pause_left = pause;
 101         do {
 102                 pause_tv.tv_sec = pause_left / NANOSEC;
 103                 pause_tv.tv_nsec = pause_left % NANOSEC;
 104                 status = nanosleep(&pause_tv, (struct timespec *)NULL);
 105                 if (status < 0) {
 105                 if (status < 0)
 106                         if (errno == EINTR) {
 107                                 now = gethrtime();
 108                                 pause_left = *wakeup - now;
 109                                 if (pause_left < 1000)
 110                                         /* Near enough */
 111                                         return;
```



```
 112                         } else {
 113                                 fail(1, "nanosleep failed");
 114                         }
 115                 }
 116 #endif /* ! codereview */
 117         } while (status != 0);
 118 }

 120 /*
 121  * Signal handler - so we can be aware of SIGCONT
 122  */
 123 void
 124 cont_handler(int sig_number)
 125 {
 126         /* Re-set the signal handler */
 127         (void) signal(sig_number, cont_handler);
 128         caught_cont = 1;
 129 }
```

```
**********************************************************
    1570 Mon Feb  3 15:34:54 2014
new/usr/src/cmd/stat/fsstat/Makefile
arcstat
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #

  26 PROG = fsstat
  27 OBJS = fsstat.o
  28 SRCS =$(OBJS:%.o=%.c) $(COMMON_SRCS)

  30 include $(SRC)/cmd/Makefile.cmd
  31 include $(SRC)/cmd/stat/Makefile.stat

  33 COMMON_OBJS = common.o timestamp.o
  34 COMMON_SRCS = $(COMMON_OBJS:%.o=$(STATCOMMONDIR)/%.c)

  36 LDLIBS += -lkstat
  37 CFLAGS += $(CCVERBOSE) -I${STATCOMMONDIR}
  38 CERRWARN += -_gcc=-Wno-parentheses
  38 FILEMODE= 0555

  40 lint := LINTFLAGS = -muxs -I$(STATCOMMONDIR)

  42 .KEEP_STATE:

  44 all: $(PROG)

  46 install: all $(ROOTPROG)

  48 $(PROG): $(OBJS) $(COMMON_OBJS)
  49         $(LINK.c) -o $(PROG) $(OBJS) $(COMMON_OBJS) $(LDLIBS)
  50         $(POST_PROCESS)

  52 %.o : $(STATCOMMONDIR)/%.c
  53         $(COMPILE.c) -o $@ $<
  54         $(POST_PROCESS_O)

  56 clean:
  57         -$(RM) $(OBJS) $(COMMON_OBJS)

  59 lint: lint_SRCS
```



```
  61 include $(SRC)/cmd/Makefile.targ
```

****
1563 Mon Feb 3 15:34:54 2014
new/usr/src/cmd/stat/iostat/Makefile
arcstat
****

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #

  26 PROG = iostat
  27 OBJS = iostat.o
  28 SRCS =$(OBJS:%.o=%.c) $(COMMON_SRCS)

  30 include $(SRC)/cmd/Makefile.cmd
  31 include $(SRC)/cmd/stat/Makefile.stat

  33 LDLIBS += -lkstat -ldevinfo -lavl
  34 CFLAGS += $(CCVERBOSE) -I${STATCOMMONDIR}
  35 CERRWARN += -_gcc=-Wno-uninitialized
  36 CERRWARN += -_gcc=-Wno-switch
  37 CERRWARN += -_gcc=-Wno-parentheses
  37 FILEMODE= 0555

  39 lint := LINTFLAGS = -muxs -I$(STATCOMMONDIR)

  41 .KEEP_STATE:

  43 all: $(PROG)

  45 install: all $(ROOTPROG)

  47 $(PROG): $(OBJS) $(COMMON_OBJS)
  48         $(LINK.c) -o $(PROG) $(OBJS) $(COMMON_OBJS) $(LDLIBS)
  49         $(POST_PROCESS)

  51 %.o : $(STATCOMMONDIR)/%.c
  52         $(COMPILE.c) -o $@ $<
  53         $(POST_PROCESS_O)

  55 clean:
  56         -$(RM) $(OBJS) $(COMMON_OBJS)

  58 lint: lint_SRCS

  60 include $(SRC)/cmd/Makefile.targ
```

```
**********************************************************
    1694 Mon Feb  3 15:34:55 2014
new/usr/src/cmd/stat/kstat/Makefile
arcstat
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #

  26 PROG = kstat
  27 OBJS = kstat.o
  28 SRCS =$(OBJS:%.o=%.c) $(COMMON_SRCS)

  30 include $(SRC)/cmd/Makefile.cmd
  31 include $(SRC)/cmd/stat/Makefile.stat

  33 LDLIBS += -lavl -lcmdutils -ldevinfo -lgen -lkstat
  34 CFLAGS += $(CCVERBOSE) -I${STATCOMMONDIR}
  35 CERRWARN += -_gcc=-Wno-uninitialized
  36 CERRWARN += -_gcc=-Wno-switch
  37 CERRWARN += -_gcc=-Wno-parentheses

  38 CPPFLAGS_sparc += -I$(SRC)/uts/sfmmu
  39 CPPFLAGS_sparc += -I$(SRC)/uts/sun4u/sunfire
  40 CPPFLAGS += $(CPPFLAGS_$(MACH))

  42 FILEMODE= 0555

  44 lint := LINTFLAGS = -muxs -I$(STATCOMMONDIR)

  46 .KEEP_STATE:

  48 all: $(PROG)

  50 install: all $(ROOTPROG)

  52 $(PROG): $(OBJS) $(COMMON_OBJS)
  53         $(LINK.c) -o $(PROG) $(OBJS) $(COMMON_OBJS) $(LDLIBS)
  54         $(POST_PROCESS)

  56 %.o : $(STATCOMMONDIR)/%.c
  57         $(COMPILE.c) -o $@ $<
  58         $(POST_PROCESS_O)

  60 clean:
```



```
  61         -$(RM) $(OBJS) $(COMMON_OBJS)

  63 lint: lint_SRCS

  65 include $(SRC)/cmd/Makefile.targ
```

****
1563 Mon Feb 3 15:34:55 2014
new/usr/src/cmd/stat/mpstat/Makefile
arcstat
****

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #

  26 PROG = mpstat
  27 OBJS = mpstat.o
  28 SRCS =$(OBJS:%.o=%.c) $(COMMON_SRCS)

  30 include $(SRC)/cmd/Makefile.cmd
  31 include $(SRC)/cmd/stat/Makefile.stat

  33 LDLIBS += -ldevinfo -lkstat -lavl
  34 CFLAGS += $(CCVERBOSE) -I${STATCOMMONDIR}
  35 CERRWARN += -_gcc=-Wno-uninitialized
  36 CERRWARN += -_gcc=-Wno-switch
  37 CERRWARN += -_gcc=-Wno-parentheses
  37 FILEMODE= 0555

  39 lint := LINTFLAGS = -muxs -I$(STATCOMMONDIR)

  41 .KEEP_STATE:

  43 all: $(PROG)

  45 install: all $(ROOTPROG)

  47 $(PROG): $(OBJS) $(COMMON_OBJS)
  48         $(LINK.c) -o $(PROG) $(OBJS) $(COMMON_OBJS) $(LDLIBS)
  49         $(POST_PROCESS)

  51 %.o : $(STATCOMMONDIR)/%.c
  52         $(COMPILE.c) -o $@ $<
  53         $(POST_PROCESS_O)

  55 clean:
  56         -$(RM) $(OBJS) $(COMMON_OBJS)

  58 lint: lint_SRCS

  60 include $(SRC)/cmd/Makefile.targ
```

****
1563 Mon Feb 3 15:34:55 2014
new/usr/src/cmd/stat/vmstat/Makefile
arcstat
****

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

PROG = vmstat
OBJS = vmstat.o
SRCS =$(OBJS:%.o=%.c) $(COMMON_SRCS)

include $(SRC)/cmd/Makefile.cmd
include $(SRC)/cmd/stat/Makefile.stat

LDLIBS += -ldevinfo -lkstat -lavl
CFLAGS += $(CCVERBOSE) -I${STATCOMMONDIR}
CERRWARN += -_gcc=-Wno-switch
CERRWARN += -_gcc=-Wno-uninitialized
CERRWARN += -_gcc=-Wno-parentheses
FILEMODE= 0555

lint := LINTFLAGS = -muxs -I$(STATCOMMONDIR)

.KEEP_STATE:

all: $(PROG)

install: all $(ROOTPROG)

$(PROG): $(OBJS) $(COMMON_OBJS)
        $(LINK.c) -o $(PROG) $(OBJS) $(COMMON_OBJS) $(LDLIBS)
        $(POST_PROCESS)

%.o : $(STATCOMMONDIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

clean:
        -$(RM) $(OBJS) $(COMMON_OBJS)

lint: lint_SRCS

include $(SRC)/cmd/Makefile.targ
```

****
487 Mon Feb 3 15:34:56 2014
new/usr/src/man/man1m/arcstat.1m
arcstat
****

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2014 David Hoeppner.  All rights reserved.
.\"
#endif /* ! codereview */
```

****
905 Mon Feb 3 15:34:56 2014
new/usr/src/pkg/manifests/system-monitoring-arcstat.mf
arcstat
****

```
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright 2014 David Hoeppner.  All rights reserved.
#

set name=pkg.fmri value=pkg:/system/monitoring/arcstat@$(PKGVERS)
set name=pkg.description \
    value=""
set name=pkg.summary \
    value=""
set name=info.classification \
    value="org.opensolaris.category.2008:System/Core"
set name=variant.arch value=$(ARCH)
dir path=usr group=sys
dir path=usr/bin
dir path=usr/share/man/man1
file path=usr/bin/arcstat mode=0555
file path=usr/share/man/man1m/arcstat.1m
license lic_CDDL license=lic_CDDL
#endif /* ! codereview */
```